MODELS AND SOLUTION APPROACHES FOR INTERMODAL AND LESS-THAN-TRUCKLOAD NETWORK DESIGN WITH LOAD CONSOLIDATIONS

A Dissertation

by

HOMARJUN AGRAHARI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2007

Major Subject: Industrial Engineering

MODELS AND SOLUTION APPROACHES FOR INTERMODAL AND LESS-THAN-TRUCKLOAD NETWORK DESIGN WITH LOAD CONSOLIDATIONS

A Dissertation

by

HOMARJUN AGRAHARI

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Halit Üster
Committee Members,	Sıla Çetinkaya
	Ricardo Gutierrez-Osuna
	Brett Peters
Head of Department,	Brett Peters

December 2007

Major Subject: Industrial Engineering

ABSTRACT

Models and Solution Approaches for Intermodal and Less-than-Truckload Network Design with Load Consolidations. (December 2007)

Homarjun Agrahari, B.Tech., Indian Institute of Technology Bombay;
M.Tech., Indian Institute of Technology Bombay

Chair of Advisory Committee: Dr. Halit Üster

Logistics and supply chain problems arising in the context of intermodal transportation and less-than-truckload (LTL) network design typically require commodities to be consolidated and shipped via the most economical route to their destinations. Traditionally, these problems have been modelled using network design or hub-and-spoke approaches. In a network design problem, one is given the network and flow requirements between the origin and destination pairs (commodities), and the objective is to route the flows over the network so as to minimize the sum of the fixed charge incurred in using arcs and routing costs. However, there are possible benefits, due to economies-of-scale in transportation, that are not addressed in standard network design models. On the other hand, hub location problems are motivated by potential economies-of-scale in transportation costs when loads are consolidated and shipped together over a completely connected hub network. However, in a hub location problem, the assignment of a node to a hub is independent of the commodities originating at, or destined to, this node. Such an indiscriminate assignment may not be suitable for all commodities originating at a particular node because of their different destinations. Problems arising in the area of LTL transportation, intermodal transportation and package routing generally have characteristics such as economies-of-scale in transportation costs in addition to the requirement of commodity-based routing. Obviously, the existing network design and hub location-based models are

not directly suitable for these applications. In this dissertation, we investigate the development of models and solution algorithms for problems in the areas of LTL and intermodal transportation as well as in the freight forwarders industry. We develop models and solution methods to address strategic, tactical and operational level decision issues and show computational results. This research provides new insights into these application areas and new solution methods therein. The solution algorithms developed here also contribute to the general area of discrete optimization, particularly for problems with similar characteristics.

To *my parents and wife*

ACKNOWLEDGMENTS

I would like to thank Dr. Halit Üster for being my advisor through the development of this dissertation and for giving me an opportunity to work as a research assistant under his supervision. I also express my gratitude to him for helping me personally and professionally, which made a difference in my stay in College Station.

I would also like to thank Dr. Sıla Çetinkaya, Dr. Ricardo Gutierrez-Osuna and Dr. Brett Peters for serving as member of my advising committee and providing valuable suggestions and comments.

I am thankful to my friends and colleagues Abhishhek Kumar Shrivastava, Dr. Balabhaskar Balasundaram and Sandeep Sachdeva who were there for support, advice, discussion or just for company. I am also thankful to my wonderful officemates: Burcu Baris Kesin, Ezgi Can Eren, Gopalakrishnan Easwaran, Hui Lin, Joaquin Torres, Liqing Zhang , Panitan Kewcharoenwong and Vishal Karnik. They made it fun to work in the lab and provided helpful suggestions and tips through discussions. I am also thankful to my friends Ashish Agrawal, Gregory Fernandes, John Vijoe, Karambir Singh Kalsi, Rajaneesh Jandhyam, Shobhit Verma, Shreesh Chandra Mishra, Vishv Jeet and Yateesh Contractor for their support and encouragement.

I want to take this opportunity to thank everyone who helped me during my tenure as the president of INFORMS student chapter. Specifically, I wish to thank Dr. Üster for being the advisor of the chapter, Dr. Peters for taking keen interest in our activities and for support through the department, Judy Meeks, Claudia Samford, Michele Bork and Katherine Edwards for administrative help, Mark Henry, Mark Hopcus and Dennis Allen for their prompt help in technical matters.

I am thankful to Texas A&M University for providing me unique learning experience and an exciting research environment. I am a proud Aggie of this esteemed university. I am also thankful to the department of Industrial and Systems Engineering for providing excellent computing and research facilities. I thank the department for providing financial support during my first two years.

I wish to thank my family: my brothers Himasnhu and Chaxwesh, my sister Nishiddha, my parents Gayatri Devi and Om Prakash without whose love and affection I could'nt have accomplished so much. I am extremely proud of my parent who have sacrificed their whole life in my upbringing. The values they instilled in me are the foundations of my achievements. It is my sincere wish that I can make them proud through services to humanity.

Finally, I can not thank my wife Vandana enough for supporting me in my decision of pursuing Ph.D. although it meant a lot of sacrifice, hardship and patience on her part. She has always been the source of my strength, courage and inspiration in completing this work. I could not have done this without her support and courage. I thank her for extending her unconditional love and for supporting this venture.

TABLE OF CONTENTS

LIST OF TABLES

TABLE Page

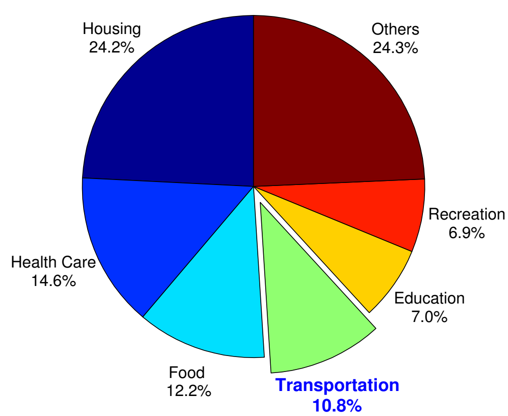
LIST OF FIGURES

CHAPTER I

INTRODUCTION

Freight transportation is the backbone of the economy. As shown in Figure 1, in 2000, transportation related services contributed more than 10% of the national GDP of the United States. Many industries own their transport operations which accounted for an additional $142 billion to the economy. In the U.S., freight transportation methods move a very large quantity of goods, e.g. in 1998, over 15 billion tons of goods worth more than $9 trillion were moved (Caldwell and Sedor, 2002). The movement of bulk goods such as grain, coal and ores, comprise a portion of the load; however, in the recent years, the percentage of consumer goods has also increased.

Figure 1 Transportation's Importance to GDP: 2000. Source: U. S. Department of Transportation, Bureau of Transportation Statistics, BTS-02-02 2002.



Freight transportation provides jobs to millions of people. In 2000, more than 10 million people were employed in transportation-related industries. Truck drivers accounted for more than 70% of the total workforce employed in freight transportation

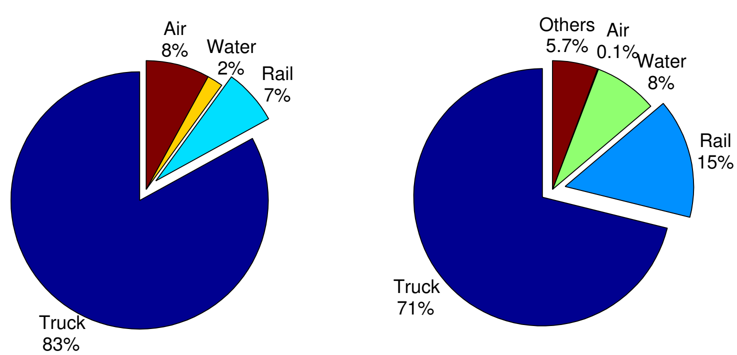
---

This dissertation follows the style and format of *Operations Research*.

related jobs. Freight transportation supports economic growth by providing service and adding value to the products transported.

The five basic forms of freight transportation are rail, highway, water, pipeline and air. The relative importance of a mode can be measured by the volume and the value of the freight transported by that mode. Figure 2 provides a summary of U.S. freight shipment by mode in 1998. It is clear from this figure that trucks transport the most freight by volume (71%) and by value (83%); in fact the volume and value transported by truck is greater than all of the other modes together. Each mode has its own characteristics that make it the mode of choice for a particular application. For example, air transportation is expensive but quick, whereas transportation by ship is slow but relatively inexpensive. Although all modes are necessary for maintaining an efficient national transportation system, clearly, trucking is the most important mode.

Figure 2 U. S. Freight Shipments by Mode: 1998, Freight Shipment in Value (L) and Tonnes (R). Source: U. S. Department of Transportation, Federal Highway Administration, 2002



I.1. Background

Safe, reliable and efficient freight transportation is essential for sustained economic growth. The savings derived from efficient logistics translate into increased investments in creating better infrastructure, manpower training, and the research and development activity needed to adapt to remaining competitive in a changing global market.

I.1.1. Intermodal and LTL Transportation

In intermodal transportation, two or more modes of transportation are used to move freight from origin to destination. Generally, the load changes mode at an intermodal terminal. It may involve multiple modes such as truck, ship, rail and air. The motivation behind intermodal transportation is to exploit economies-of-scale inherent in the modes involved, and thereby, offer the service at least total cost. For example, TOFC (a term used in intermodal transportation that stands for trailer on a flatcar), combines the flexibility of a truck for short distances where rail does not reach with rail for longer distances. Efficient and safe intermodal transportation requires specialized equipment for handling and an intermodal terminal for mode change. Similarly, containerships are an intermodal type of transport that utilizes waterways where the whole container or railcar is loaded onto a ship to exploit economy-of-scale of waterways transport. Similarly air-truck is a common choice of intermodal transportation mostly for applications that require faster service. In summary, intermodal transportation results from exploiting the economies-of-scale of multiple mode, and it provides a flexible option to the logistics planner (Bowersox et al., 2002).

In the trucking industry, two main types of services are available: truckload (TL) and less-than-truckload (LTL). In TL transportation, a shipper hires a full truck to ship a consignment from its origin to its destination. The shipper must pay for the full truck capacity irrespective of the capacity utilized by the consignment. The truck does not make stops enroute to load or unload additional loads. TL transportation is relatively faster and more reliable because of no stops in between origin and destination. In contrast, LTL is a service for shippers that need only a small quantity of goods delivered. A LTL shipment is delivered with various other shipments and is usually not delivered directly to a destination. The cost of less-than-truckload shipments are less relatively, but since the shipment may make multiple stops before reaching the final destination, delay may result.

Railroads carry 15% of US freight by value and 7% by volume. Railroads are generally suitable for large tonnage over long distance. The main commodities transported by rail includes of ores, coal, automobiles, and farm equipment. In the early nineteen seventies, railroads started to become a serious player in intermodal and container traffic. From 1996 to 2006, the rail intermodal share has increased at a rate of 55.5%. In today's increasingly global economy, railroads are playing an important role in the transportation structure as a intermodal transportation leader (Bowersox et al., 2002).

In summary, intermodal and LTL transportation are most important modes for freight transportation today and their importance can be realized by the quantity of freight they carry each year, their contribution to economic wellbeing of the nation in terms of employment, value addition to the goods, and employment creation.

I.1.2. Consolidation

In freight transportation, the transportation rates generally are specified per hundredweight, and it is a general rule that the larger the shipment, the lower the cost per hundredweight. Quantity discounts are present in many businesses because of better capacity utilization. In internodal and LTL transportation, it is often the case that the individual load is not large enough to qualify for quantity discount published in common carrier transportation rates (Bowersox et al., 1986; Ghiani et al., 2004). Load consolidation or consolidated transportation is a practical tool for combining loads to exploit the benefits of economies-of-scale and reducing total transportation costs. The following three types of load consolidation are common.

1. *Market Area Consolidation:* This type of consolidation involves grouping loads from customers based on their geographical location and it does not break the natural flow by changing timing, etc. Variation in daily volume poses a problem for such consolidation as there may not be enough load in the market area on a particular day. This problem is overcome by either break-bulk or holding the consolidation until the scheduled delivery. Another option is to pool delivery from a third party logistics firm.

2. *Scheduled Delivery:* This strategy implies delivery on a fixed schedule in the anticipation of desired load consolidation.Railroad are a good example for scheduled delivery.

3. *Pooled Delivery:* Participation in a pooled delivery typically means to utilize service from a third party provider. This form of delivery is dependent upon third party and the load may wait for long time while waiting for the the third party, which may be long, but generally it is shorter than the wait without pooling the delivery with third party.

We refer the reader to Bowersox et al. (2002) for detailed discussion on consolidation.

I.1.3. Motivation and Scope of the Dissertation

Due to multiple factors such as globalization, economic growth, and increased demand, in recent years freight transportation has changed in many ways. In the light of some facts presented above, we summarize the observations that motivated this dissertation research topic.

1. Freight transportation has increased dramatically with population and economic growth. Increase in the population has led to an increase in demand, and globalization has contributed to greater interdependence of economies across the globe (FWHA, 1998).

2. The average cost for freight has decreased from 16.1% of the GDP in 1984 to 10% of the GDP in 2000. This improvement in productivity can be attributed to deregulation, investment in infrastructure, technology and the adoption of more efficient strategic and operational practices. Higher productivity in freight transportation means that industries are better able to compete in the global economy (FHWA, 2006b).

3. Today, commitment to service and transit times is of unprecedented importance. Customers demand more flexible, reliable and timely service. The penalty for not meeting service quality levels is severe. Issues of reliability, timely service and service quality are forcing a shift from hub-and-spoke to point-to-point delivery.

4. Trucking and intermodal transportation together move more than 85% of the total US freight business. Trucking is the most popular mode accounting for about 78 to 79% of the load, whereas the railroad moves 8% of the nation's freight bill.

> With the increased interdependence of economies, intermodal transportation is projected to increase in the future (FWHA, 1998; FHWA, 2006a).

While the facts above show that transportation is important and intermodal and LTL transportation industries are important for the economy, it also underlines the immediate need of making the operations more cost efficient in order for the firms to remain competitive in today's global economy. In an increasingly competitive market with shrinking margins, firms are getting increasingly interested in load consolidation to reduce transportation costs. Consolidation, however, requires intermediate handling and delays. Therefore, the challenge is to reduce costs by consolidation, and at the same time, to maintain high service quality. In applications such as intermodal and LTL transportation, en-route handling, loading, unloading and sorting may nullify the benefits of consolidation and economies-of-scale, and cause unnecessary delays. Therefore, LTL and intermodal transportation industry is shifting from hub-and-spoke to point-to-point delivery system. Since this area is new, quantitative models and solution methods are not abundant. As observed by Crainic and Kim (2005), intermodal transportation is a young area and current models are comparatively new. The hub-and-spoke network models are not suitable for addressing the need of dedicated models for network design problems in the context of intermodal transportation and LTL transportation problems. The traditional models of LTL and intermodal transportation use hub-and-spoke and network design type approaches. These lack one or more of the following:

1. Commodity-based routing decisions.
2. Explicit consideration of economies of consolidation.
3. A special network structure with consolidation and deconsolidation centers.

4. Single sourcing constraints that force the commodity to flow on a single path in order to avoid unnecessary operational complexities and delays.

It has become important for freight transportation businesses to develop strategic plans and achieve higher operational efficiency. This dissertation addresses strategic and tactical as well as operational level problems to provide a complete solution.

In the next section we define the network in the context of intermodal transportation and LTL transportation, followed by an auxiliary network that is an abstract representation of the actual network and it helps visualize the operations that we consider in our problems.

I.2. Auxiliary Network Representation

In intermodal and LTL network design, we are given a network with multi-commodity flows where each commodity is defined by its unique pair of origin and destination nodes and a known required flow amount. The system is operated in such a way that the commodities are collected and consolidated into truckloads at consolidation centers, a linehaul transfer takes place for the consolidated loads, which are deconsolidated at deconsolidation centers and from there, the commodities are shipped to their final destinations. Notice that operational characteristics are different than hub location because the consolidation activity is based on the commodity and not the physical nodes, and different than traditional network design because it considers consolidation explicitly and also restricts the flow to use maximum of three arcs (collection,linehaul transfer, distribution).

Considering a general physical network (e.g., a road network), we observe that the underlying graph of a multi-commodity flow network under the above described operational activities (consolidation, linehaul and deconsolidation) can be conceptualized as three directed graphs concatenated with common node sets to form a 3-part network where the parts correspond to the interactions within each activity. This 3-part network is an "auxiliary" network, and it provides an abstraction of the general physical network for our problem. In order to build the auxiliary network, we define the following sets: $\mathcal{P} = \{p_1, \ldots, p_N\}$ is the set of required commodity flows. $\mathcal{F}=\{f_{p_1}, \ldots, f_{p_N}\}$ and $\mathcal{T}=\{t_{p_1}, \ldots, t_{p_N}\}$ represent the sets of nodes representing commodity origins and destinations, respectively. The sets $\mathcal{J}$ and $\mathcal{K}$ are the physical nodes where the consolidation and deconsolidation activities take place, respectively. To build the auxiliary network $G = (V, A)$, we introduce three sets of arcs. A_{FJ}, A_{JK} and A_{KT} are the sets of all arcs between $\mathcal{F}$ and $\mathcal{J}$, $\mathcal{J}$ and $\mathcal{K}$, and $\mathcal{K}$ and $\mathcal{T}$, respectively. The length of an arc either corresponds to the shortest possible path length represented by this arc on the physical network, or it is specified according to possible shipment routes. Subsequently, we define three directed bipartite graphs which we concatenate with overlapping common nodes to form the auxiliary network. The first such graph is $G^C(\mathcal{F} \cup \mathcal{J}, A_{FJ})$; the second is $G^L(\mathcal{J} \cup \mathcal{K}, A_{JK})$; and the third is $G^D(\mathcal{K} \cup \mathcal{T}, A_{KT})$. Thus, we have $V = \mathcal{F} \cup \mathcal{J} \cup \mathcal{K} \cup \mathcal{T}$ and $A = A_{FJ} \cup A_{JK} \cup A_{KT}$. Furthermore, the auxiliary network also includes the set of arcs $(f_{p_i}, t_{p_i}),\ \forall\, p_i \in \mathcal{P}$, which represents possible direct shipments. Note that the nodes in the set $\mathcal{F}$ are not necessarily distinct because one node may be origin of more than one commodities therefore appearing more than once in the set $\mathcal{F}$. Same holds true for the nodes in set $\mathcal{T}$. Since a node which is origin for a commodity may be the destination of another commodity, it is easy to see that the sets $\mathcal{F}$ and $\mathcal{T}$ may overlap. Similarly, since a node in physical network may be candidate for consolidation center as well as deconsolidation center, the sets $\mathcal{J}$ and $\mathcal{K}$ may also overlap.

An important characteristic of operations in our problem is that, due to the consideration of possible consolidation and deconsolidation locations on the physical network, sets $\mathcal{J}$ and $\mathcal{K}$, when a commodity is assigned to a consolidation center, which refers to a node on the physical network, that center and the *commodity's physical origin node* may coincide. The same situation applies for the destinations as well. This property implies that commodities originating at the same physical node can be *consolidated at their source*. Similarly, commodities destined to the same physical node can be *deconsolidated at their destination*.

Another important characteristic is simple yet effective consideration of transportation *economies-of-scale* realized through TL consolidations using capacitated trucks. In addition, any fixed costs associated with TL shipments can also be easily incorporated.

For the linehaul TL shipments between the regional centers, we assume that the travel follows the shortest path on the physical network or that it is specified according to possible shipment routes.

I.3. Problem Description

In intermodal and LTL transportation, a company must make the following types of decisions.

1. Strategic decisions about capital investments such as buildings, facilities and equipment for loading, unloading and sorting activities. Transportation capacities must be acquired in the form of either owned, rented or a combination of both.

2. Tactical capacity planning decisions regarding the allocation of trucks on links and related equipment from either own or third party arrangement. Since the cost of acquiring capacity on expedited basis is much more than the regular price; therefore, appropriate capacity planning can help reduce the cost of emer-

gency capacity acquisition. Other tactical decisions may include planning for human resources.

3. Operational decisions such as truck-linehaul assignments and commodity-truck-linehaul assignments as well as other operational constraints.

Obviously, the decision problems at the strategic, tactical and operational levels are inter-related. Decisions made at the strategic level provide the resources/information required at the tactical and operational level, and, similarly, decisions made at the strategic and tactical levels affect operational level decisions.

We first define the problem on a general network and then create an auxiliary network to develop a corresponding mathematical formulation for the problem. This abstraction to an auxiliary network, as will become clear, facilitates the development of models that explicitly capture economies-of-scale and commodity origin and destination-based routing requirements. We provide formal descriptions of the problems that address these three decision levels below.

I.3.1. Strategic Network Design Problem (SNDP)

In SNDP, we have a network with multi-commodity flows where each commodity is defined by its unique pair of origin and destination nodes and a known required flow amount. The system is operated in such a way that the commodities are collected and consolidated into truckloads at consolidation centers, and then a linehaul transfer takes place for the consolidated loads, which are deconsolidated at deconsolidation centers and from there, the commodities are shipped to their final destinations. In order to design a network assuming such operation, several decisions must be made, which include 1) the locations and capacities of the consolidation and deconsolidation centers; 2) the linehaul transfer links and their capacities in terms of the number of *truckload trips* between the consolidation and deconsolidation centers, and 3) the assignment of commodities to consolidation and deconsolidation centers and, in turn, to transfer links. The costs in the system include collection costs, linehaul transfer costs, distribution costs, and costs for locating consolidation and deconsolidation centers. We assume that the capacity installments on linehaul transfer links and at the consolidation and deconsolidation centers are set in fixed increments. On transfer links, capacity can be installed with increments of truckload capacity and at centers, with increments of some base capacity, both with their associated incremental costs.

I.3.2. Tactical Network Design Problem (TNDP)

In TNDP, we consider a network with multi-commodities as described in SNDP. The decisions to be made are 1) the connections and capacities in terms of the number of *truckload trips* between the consolidation and deconsolidation centers, and 2) the assignment of commodities to consolidation and deconsolidation centers and, in turn, to transfer links. The costs in the system include collection costs, linehaul transfer costs and distribution costs. We assume that capacity installments on linehaul transfer links can be set in fixed increments of truckload capacity with associated increment costs. Again, we assume that a commodity can follow only a single route from origin to destination. Thus essentially, if we fix the center locations in SNDP and do not consider capacity limitations on them, we obtain TNDP.

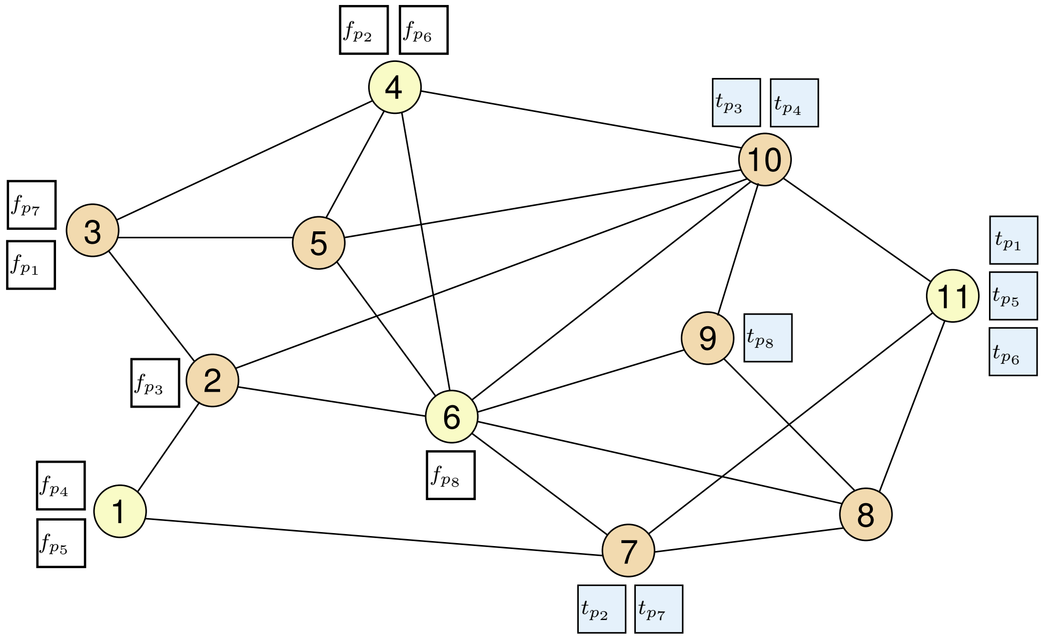
I.3.3. Operational Network Design Problem (ONDP)

In ONDP, we have a fleet of trucks in addition to the network and commodity flows. The system is operated similarly to the one described previously, i.e. consolidation, linehaul and distribution. Additionally, we allow direct shipments between origin and destination nodes since this is preferred when the origin and destination nodes of a commodity are relatively close, and, thus, consolidation does not make economical sense. The decisions to be made in ONDP include 1) the assignment of trucks to linehaul transfer links 2) the assignment of commodities to a truckload shipment established on transfer links 3) the identification of commodities that are to be shipped directly.

I.3.4. An Example

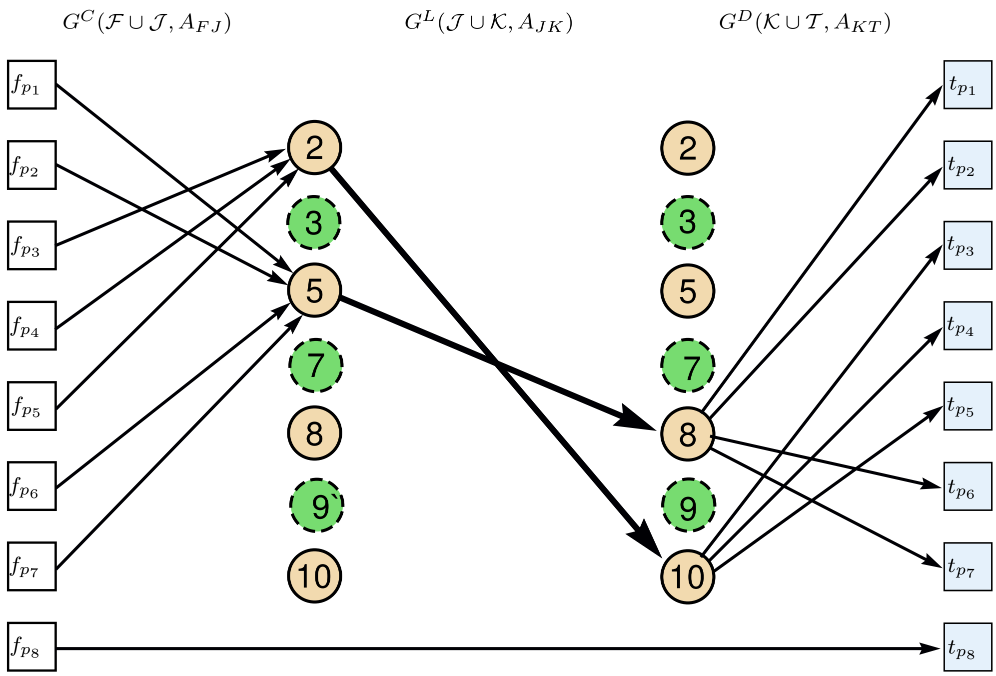
For illustration purposes, consider a general physical network with 11 nodes of which seven (nodes 2, 3, 5, 7, 8, 9 and 10) are candidate centers for consolidation and deconsolidation depicted by shaded circles in Figure 3. The origins and destinations of eight commodities are also shown on this network. An example flow of commodities on the corresponding auxiliary graph is shown in Figure 4. In the context of SNDP, this example implies that 1) out of the seven possible candidates for consolidation and deconsolidation, four nodes (2, 5, 8 and 10) have been selected depicted by solid circles (and capacities installed on them), and nodes 3, 7 and 9, shown by dashed circles, are not selected; 2) connections and capacities in terms of *truckload trips* are depicted by solid arrows 2-10 and 5-8; 3) commodities p_1, p_2, p_6 and p_7 are assigned to collection and distribution centers 5 and 8, respectively, commodities p_3, p_4 and p_5 are assigned to collection and distribution centers 2 and 10, respectively (which, in turn implies their assignment to links 2-10 and 5-8, respectively). In the context of TNDP, this example solution implies capacities on links 2-10 and 5-8, and the assignment of commodities to linehaul links 2-10 and 5-8 similarly to the method described for SNDP. In the context of ONDP, the solution implies that commodities p_1, p_2, p_6 and p_7 are shipped via LTL mode to the center at node 5 for consolidation into a TL shipment that is transferred over a linehaul link to the center at node 8 for deconsolidation; from there, the individual commodities are shipped to their final

Figure 3 Required Commodity Flows on a Physical Network



destinations. Similarly, commodities p_3, p_4 and p_5 are consolidated into a TL shipment and transferred from the consolidation center at node 2 to the deconsolidation center at node 10. Finally, commodity p_8 is sent via direct shipment, which can be justified by the short distance between f_{p_8} and t_{p_8}, as shown in Figure 3. Note that, in the case of ONDP, it is assumed that the total commodity flow assigned for TL shipments is less than the *individual* truck capacity whereas in case of TNDP and SNDP, we are only concerned with aggregated capacities i.e., aggregate capacity on each link should be more than the load assigned on that link. In the context of SNDP, the shaded circles with dotted outline as shown in Figure 4 represent the candidates centers for consolidation and deconsolidation, that were not located in the example solution. ONDP and TNDP do not consider center location, therefore shaded circles are relevant only in the context of SNDP.

Figure 4 An Example Routing in the Auxiliary Network



I.4. Experiment Data and Computational Study

Linehaul consolidated shipments occur more realistically between largely separated geographical regions; for example, shipments between eastern and western regions of the continental U.S. or between northern and southern parts of the eastern (or western or central) U.S. To represent such transportation characteristics in our randomly generated problem instances, we create two identical squares of size E separated by a certain fixed distance A (square center-to-square center) horizontally. The left square consists of only origin nodes and the right square consists of only destination nodes. We generate an equal number N_P of uniformly distributed point coordinates in each of the squares representing the physical origin and destination nodes. Then, we randomly select N distinct pairs of these physical origin and destination nodes to determine N commodities. For each commodity i, we generate its required commodity flow w_i units randomly using Uniform$[0.2, 0.8]$. We also select M distinct nodes from each square to be designated as centers. We employ the Euclidean norm to calculate the distances. The network generation approach described so far is similar to the approach used in other LTL network design studies including Amiri and Pirkul (1997); Balakrishnan and Graves (1989); Muriel and Munshi (2003).

Since the collection process from origin nodes to consolidation centers and the distribution process from deconsolidation centers to final destinations involve smaller loads, we consider the case where the shipment at these stages use the LTL (Less-Than-Truckload) mode of transportation, and thus, the costs are based on the load amount and the distance. Typically, a constant dollar value per unit per mile of shipment is used to represent LTL type costs. Note that this also applies to direct shipments. On the other hand, since consolidated larger loads (truckloads) are transferred between consolidation and deconsolidation centers, we consider the case where the shipments at this intermediate stage utilize the TL (TruckLoad) mode, thus, incurring per mile full TL costs for this stage. In this case, a constant per mile dollar value for dispatching a truck is used to calculate the total cost of a shipment between two centers. In terms of cost parameters, we take LTL type costs, α^f, α^t and α^{ft}, as 1.0, 1.0 and 1.2 per unit per mile, respectively. Since TL type cost β depends upon the capacity of the truck, to select the β, we use the criterion that a TL shipment becomes economical for loads that are equal to or more than 75% of the truck capacity, i.e., β is given by $0.75 \times U \times \alpha^{ft}$.

All of the computational studies were performed on a machine with Pentium D 3.2 GHz CPU with 1.0 GB RAM. The algorithms were implemented using C++ utilizing STL (Standard Template Library) and Concert Technology when CPLEX was used. Note that, although an attempt was made to write efficient code, the purpose of comparing the developed algorithms with commercial grade CPLEX is to show the efficiency of the algorithms not the implementation. The author does not claim the code is written as efficiently as commercial grade software, and therefore, does not rule out the possibility that the performance of the implementation could be improved to a certain extent.

I.5. Organization of the Dissertation

The dissertation is organized as follows. In Chapter II, we present a brief overview of the relevant literature on hub location, network design, facility location and service network design problems. In Chapter III, we develop a model and heuristic solution algorithms for ONDP, which is a multicommodity flow network design problem with capacity installments. Chapter IV focuses on a tactical level problem TNDP, which is a single source capacitated facility location problem with staircase capacities. We present a model and algorithms for finding the lower and upper bounds for the TNDP. The algorithmic framework developed in solving the ONDP in Chapter III is utilized to develop a Lagrangian Heuristic solution method for finding the upper bounds for TNDP. In Chapter V, we focus on the strategic network design problem, which now also includes location and capacity decisions about the consolidation and deconsolidation centers apart from the design of the transportation network. We develop mathematical model and Benders decomposition-based solution for the SNDP. Finally, in Chapter VI, we conclude with a brief summary of the research results and a discussion of the potential impact of this dissertation in the future.

CHAPTER II

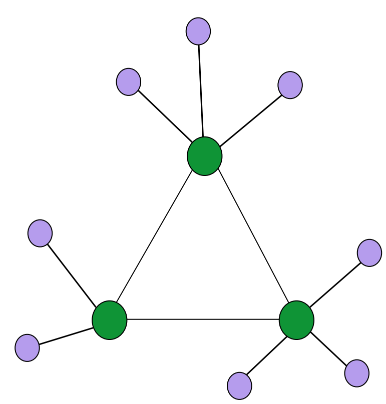
LITERATURE REVIEW

Since our problem concerns the design of a distribution network and involves consolidation and deconsolidation activities, it naturally relates to the hub location literature. Since we are interested in routing commodities over a network, the general area of network design is also relevant. Furthermore, due to the specific application area, studies in logistics service network design and load planning are also within our scope. Finally, since each linehaul link can also be considered as a facility, our problem is related to capacitated facility location problems. Below, we provide a rather macroscopic literature review highlighting and discussing the content in each relevant area.

II.1. Hub Location

Hub location problems are concerned with locating hubs and assigning the nodes of a physical network to these hubs in such a way that the total cost of the fixed hub locations and transportation are minimized over the network. An example hub location

Figure 5 Hub Location Problem



problem is shown in Figure 5. In Figure 5, small circles represent clients that need to be assigned to hubs indicated by large circles. Notice that the subgraph induced by the hubs is complete. Generally, the transportation cost for inter-hub transfer is discounted (due to the economies-of-scale for inter-hub transfers) which provides the motivation for locating hubs. "Hub" is a general term used to refer to location or a point where a commodity or information from several sources gets consolidated to go either to another hub or to its final destination. Common examples are hub and spoke network in air transportation, LTL transportation, and telecommunications. We refer the reader to Campbell (1994), Campbell et al. (2002) and O'Kelly and Miller (1994) for exhaustive surveys of the hub location literature.

There are several variants of hub location problems, such as the single/multiple allocation hub location problem, the capacitated or uncapacitated hub location problem, and the p-hub location problem. Also, there may, or may not, be a fixed charge for locating the hubs. We refer the reader to O'Kelly and Miller (1994) who develop a hub network classification system based on characteristics of service nodes, hubs and arcs.

Different models are motivated by different applications, and below we present a generic model for a capacitated single allocation hub location problem adapted from Ernst and Krishnamoorthy (1999) with some notational changes.

II.1.1. The Model

We will use the notation given below to describe the mathematical formulation. Let $\mathcal{I} = \{1, \ldots, n\}$ be the set of nodes and $\mathcal{H} = \{1, \ldots, m\}$ be the set of candidate hub locations in the network.

Decision Variables:

z_{ij} 1 if a node i is allocated to the hub located at node j,0 o.w.

z_{kk} 1 if a node k is selected as a hub, 0 o.w.

y^i_{kl} total flow emanating from node i routed between hubs k and l.

Parameters:

w_{ij} flow between nodes i and j.

d_{ij} the distance between nodes i and j.

O_i $\sum_{j\in\mathcal{I}} w_{ij}$

D_i $\sum_{j\in\mathcal{I}} w_{ji}$

f_k fixed cost of locating hub at k.

b_k capacity of hub k.

Objective and Constraints:

$$\text{Min} \quad \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{H}} d_{ik}z_{ik}(\chi O_i + \delta D_i) + \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{H}}\sum_{l\in\mathcal{H}} \alpha d_{kl} y^i_{kl} + \sum_{k\in\mathcal{H}} f_k z_{kk} \tag{2.1}$$

subject to

$$\sum_{k\in\mathcal{H}} z_{ik} = 1 \qquad \forall\, i \in \mathcal{I} \tag{2.2}$$

$$z_{ik} \le z_{kk} \qquad \forall i \in \mathcal{I}\,, k \in \mathcal{H} \tag{2.3}$$

$$\sum_{k\in\mathcal{H}} y^i_{kl} - \sum_{l\in\mathcal{H}} y^i_{kl} = O_i z_{ik} - \sum_{j\in\mathcal{I}} W_{ij} z_{ik} \qquad \forall i \in \mathcal{I}\,, k \in \mathcal{H} \tag{2.4}$$

$$\sum_{i\in\mathcal{I}} O_i z_{ik} \le b_k z_{kk} \qquad \forall k \in \mathcal{H} \tag{2.5}$$

$$z_{ik} \in \{0,1\}^n \qquad \forall i \in \mathcal{I}\,, k \in \mathcal{H} \tag{2.6}$$

$$y^i_{kl} \ge 0 \qquad \forall i \in \mathcal{I} \text{ and } k,l \in \mathcal{H} \tag{2.7}$$

In the objective function given by expression (2.1), the first term represents the total transportation cost for the collection and distribution operations and the second and third terms represent the total transportation cost for the inter-hub transfers and the fixed cost of locating the hubs, respectively. Constraint set (2.2) ensures that each commodity is assigned to exactly one hub. Constraints set (2.3) ensures that a hub is located if the node is assigned to itself, and constraint set (2.4) ensures flow conservation at the hubs and constraint set (2.5) implies capacity restriction. Constraint sets (2.6) and (2.7) impose standard binary restrictions and non negativity restrictions on decision variables $\mathbf{z}$ and $\mathbf{y}$ respectively. The parameters χ, δ, α represent the per unit per mile cost of transportation for collection, distribution and inter-hub transfer. Generally, the inter-hub costs are discounted, i.e. $\alpha < \chi$ and $\alpha < \delta$. Depending upon the application, there may be restriction on the maximum number of hubs a commodity can visit before it reaches its destination. There are several alternate formulations for various hub location problems and we refer the reader to Campbell (1994) for details. We do not cover the p-Hub median problem in which the number of hubs to be located is pre-specified. However, it can be included in the formulation presented above.

Among the several hub location problems, uncapacitated and capacitated hub location problems have received more attention from the research community and are also relevant to our problem. Hub location problems are in general very difficult to solve, significantly more difficult than the classical facility location problem. As observed by Campbell (1994), even a 50 node and 5 hub problem can pose a significant solution challenge. Further the capacitated models are significantly more challenging than the uncapacitated ones. Although it is possible to obtain exact solutions to small problems, researchers have turned to heuristics for the solution of large size problems.

Traditional hub location models assume a complete subgraph formed by arcs between the hubs. According to Campbell et al. (2005a), this assumption imposes a topological and cost structure that may not be desired or realistic in many settings such as LTL network design. They propose a new model called the hub arc location model to overcome the restrictions due to the assumptions. In the first part, they introduce this new model, and examine four special cases in detail. In a companion paper, Campbell et al. (2005b) provide an integer programming formulation for hub arc problems and solution algorithms.

Our problems TNDP and ONDP, however, do not involve hub (or center) location decisions. Further, in all three levels of problems ONDP, TNDP and SNDP- we consider explicit commodity-based routing decisions, which is not the case in hub location problems. More specifically, we are interested in assigning commodities to consolidation and deconsolidation centers as opposed to hub location problems where the assignment of a node to a hub implicitly determines the assignment of the commodities originating from, or destined to, a particular node to that hub. Third, we explicitly consider capacity issues on transfer links by incorporating the assignment of capacitated trucks to linehaul links and the assignment of commodities to these trucks. Finally, in hub location models, the transportation costs between hubs typically involve the use of discounted per unit per mile costs. In our case, we consider consolidation into TL shipments and the associated costs explicitly. Therefore, despite operational similarities, the problems considered in this dissertation have fundamental differences from the hub location problems, and these differences hinder the efficient use of the modelling and solution approaches devised for these problems.

II.2. Fixed Charge Network Design

Another area related to our problem falls under the general heading of multi-commodity flow network design problems or freight transportation problems.

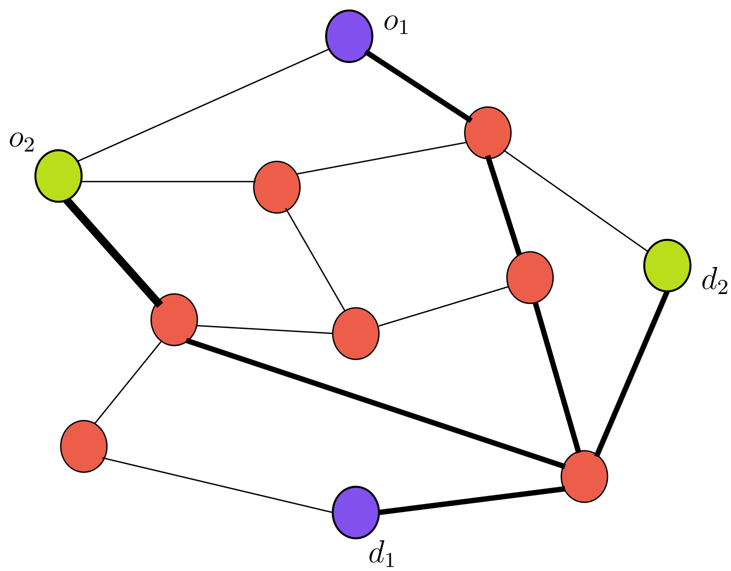


Figure 6 Fixed Charge Network Design Problem

In the general network design problem, given a set of nodes, a set of arcs between nodes, and the flow required from origin to destination (as shown in Figure 6), the problem is to design the network so as to route the flows at minimum cost. There are many variations of this problem, such as directed or undirected and capacitated or uncapacitated arcs. In the example shown in the Figure 6, commodity 1 is originating at a node labelled as o_1 and destined to node d_1 and commodity 2 is originating at a node labelled as o_2 and destined to node d_2. In the example solution, arcs shown by solid bold lines suggest the designed network for routing the commodities.

Multi-commodity flow network design models, or freight transportation studies, generally began to appear in the 80's, after the deregulation of transportation services in the United States when a severely competitive environment forced the common-carriage companies to search for ways to improve the efficiency of their operations. Early studies (Powell and Sheffi, 1983; Magnanti and Wong, 1984; Crainic and Rousseau, 1986; Powell, 1986; Delorme et al., 1987; Lamar and Sheffi, 1987; Powell and Sheffi, 1989; Braklow et al., 1992) were mostly practical cases that provided theoretical insight and an introduction to the general network design area, generally in the context of freight transportation. We note that almost all of the studies in this area are concentrated on problems related to the carrier's perspective and heavily focused on the routing of the flow on a given system (network loading) (Crainic, 2000). There are two mainstream equivalent formulations for network design/arc loading models : *arc-based* and *path-based*. In the former, the decisions relate to which arcs should be created and how the commodities should be assigned to these arcs. In most of these studies, commodities can be bifurcated so that portions of the commodity can follow different routes from origin to destination. In the latter, all of the possible paths for the commodities are considered as decision variables. There are primarily two classes of problems utilizing these formulations. *Uncapacitated* network design models concern the routing of multi-commodity flows over general networks where there are fixed charges associated with each arc (Ahuja et al., 1993; Balakrishnan et al., 1989; Barnhart et al., 2000; Crainic, 2000; Crainic and Laporte, 1997; Eckstein and Sheffi, 1987; Farvolden and Powell, 1994; Lamar et al., 1990; Rardin and Choe, 1979; Rardin, 1982). Concave arc cost models are considered in some studies (Amiri and Pirkul, 1997; Balakrishnan and Graves, 1989; Croxton et al., 2003b; Minoux, 1989). *Capacitated* models, attracted the attention of researchers only recently (Agarwal, 2002; Atamtürk, 2002a,b; Crainic et al., 2000; Magnanti et al., 1993, 1995). These works are usually motivated by telecommunication applications where communication lines with different capacities must be considered in designing the networks. Recent reviews of general network design and the freight transportation area can be found in (Crainic, 1999, 2000; Crainic and Laporte, 1997; Gendron et al., 1998; Minoux, 2001); also see Dell'Amico et al. (1997) for an extensive bibliography. We note that, with the exception of a few studies found in the network design literature (Crainic et al., 1990; Guelat et al., 1990), almost all of the other studies are related to tactical planning or minimum cost flow problems.

II.2.1. The Models

The network design problem can be formally described as follows. In the general network design problem, given a set of nodes, a set of arcs between nodes, and the flow required from origin to destination, the problem is to design the network to route the flows at minimum cost. We use the formulation provided in Crainic (1999). Given a network $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is a set of nodes, and $\mathcal{A}$ a set of arcs and let $\mathcal{P}$ be the set of commodities and let i, j be the node indices, p the product index, and let (i, j) represent an arc between nodes i and j. There are two main variations of formulations used to model the network design problem, arc-based and path-based formulation. Rardin and Choe (1979) compared arc-based and path-based models and they found out that in capacitated case, none of them has stronger formulation than the other, however in uncapacitated case, arc-based formulation provides better LP-relaxation. Below we describe both models.

Arc Based Formulation

Parameters:

f_{ij} fixed cost of using the arc i, j

c_{ij}^p transportation cost per unit per mile for commodity p on arc (i, j)

u^{ij} capacity of arc (i, j)

d_i^p demand of product p at node i.

Decision Variables:

y_{ij} 1 if arc (i, j) is used, 0 otherwise,

x_{ij}^p amount of flow of commodity p using arc (i, j)

Objective and Constraints:

$$\text{Min} \quad \sum_{(i,j)\in\mathcal{A}} \sum_{p\in\mathcal{P}} c_{ij}^p x_{ij}^p + \sum_{(i,j)\in\mathcal{A}} f_{ij} y_{ij} \tag{2.8}$$

subject to

$$\sum_{j\in\mathcal{N}} x_{ij}^p - \sum_{j\in\mathcal{N}} x_{ji}^p = d_i^p \qquad \forall i \in \mathcal{N}, p \in \mathcal{P}. \tag{2.9}$$

$$\sum_{i\in\mathcal{P}} x_{ij}^p \leq u_{ij}\, y_{ij} \qquad \forall (i, j) \in \mathcal{A}. \tag{2.10}$$

$$y_{jk} \in \mathcal{Z}^+, \text{ and } x_{ij}^p \geq 0 \qquad \forall p \in \mathcal{P}, (i, j) \in \mathcal{A}. \tag{2.11}$$

Let w^p be the total demand for the product p; then d_i^p is given by:

$$d_i^p = \begin{cases} w^p & \text{if node } i \text{ is the origin of commodity } p, \\ -w^p & \text{if node } i \text{ is the destination of commodity } p, \\ 0 & \text{otherwise.} \end{cases}$$

In the objective function given by expression (2.8), the first term represents the total flow cost and the second term represents the total fixed cost of opening fixed-charge arcs. Constraint set (2.9) ensures that each commodity is shipped from its origin to its destination. Constraint set (2.10) ensures that for each arc (j, k), the total weight of the commodities assigned does not exceed the capacity installed on that link. Constraint set (2.11) imposes an integrality restriction on decision variables $\mathbf{y}$ and a nonnegativity restriction on decision variables $\mathbf{x}$.

Path Based Formulation

The equivalent path based formulation can also be developed as follows.

Parameters:

$\mathcal{L}^p$ set of paths for commodity p,

h_p^l flow of commodities p on path l,

k_p^l transportation cost of commodity p on path l given by $\sum_{(i,j)\in\mathcal{A}} c_{ij}^p \delta_{ij}^{lp}$,

Decision Variables:

y_{ij} 1 if arc (i, j) is used, 0 otherwise,

δ_{ij}^{lp} 1, if arc (i,j) belongs to path l for commodity p, 0 otherwise

Then, the problem can be formulated as follows:

$$\text{Min} \sum_{l\in\mathcal{L}} \sum_{p\in\mathcal{P}} k_l^p h_l^p + \sum_{ij\in\mathcal{A}} f_{ij} y_{ij} \tag{2.12}$$

subject to

$$\sum_{l \in \mathcal{L}^p} h_l^p = w^p \qquad \forall p \in \mathcal{P}. \tag{2.13}$$

$$\sum_{l \in \mathcal{L}} \sum_{p \in \mathcal{P}} h_l^p \delta_{ij}^{lp} \leq u_{ij}\, y_{ij}, \qquad \forall (i,j) \in \mathcal{A}. \tag{2.14}$$

$$y_{jk} \in \mathcal{Z}^+, \qquad \forall (i,j) \in \mathcal{A} \tag{2.15}$$

$$h_l^p \geq 0, \qquad \forall p \in \mathcal{P}, l \in \mathcal{L}^p \tag{2.16}$$

In the objective function given by expression (2.12), the first term represents the total flow cost and the second term represents the total fixed cost of opening fixed-charge arcs. Constraint set (2.13) ensures that each commodity is shipped from its origin to its destination. Constraint set (2.14) ensures that, for each link (j, k), the total weight of the commodities assigned does not exceed the capacity installed on that link. The constraint sets (2.15) and (2.16) impose an integrality restriction on decision variable $\mathbf{y}$ and a nonnegativity restriction on decision variables $\mathbf{h}$, respectively.

II.2.2. Solution Methods

There are efficient procedures for solving the uncapacitated problem; the capacitated case, however, poses algorithmic and solution challenges. The presence of capacities makes the problem more difficult to solve, and sometimes even obtaining a feasible solution is a significant challenge (Gendron et al., 1998). The research approaches can be categorized into simplex based cutting plane methods, Lagrangian relaxation , Benders decomposition, and heuristics.

Each approach has its own advantages and disadvantages. The simplex based approaches can utilize the efficient codes that are now available for solving LPs and hence provide good lower bounds. However, simplex based approaches do not exploit the special network structures that might be useful, and sometimes the LPs might be highly degenerate (Bienstock and Günlük, 1996). Moreover as the LP becomes large, memory becomes a bottleneck in solving real life problems. On the other hand, Lagrangian approach exploits the problem structure and facilitates heuristic design but sometimes it is very difficult to solve the lagrangian dual. Lemaréchal (1989) and Rardin (1982) used dual ascent and sub-gradient optimization to solve the resulting lagrangian dual. Their results are shown for only the uncapacitated case, however an excellent comparison of the strengths of the various lagrangian relaxations for capacitated problems is presented in Gendron et al. (1998) where they also describe an efficient procedure based on resource-decomposition principles for identifying feasible good quality solutions. Holmberg and Yuan (2000) obtained good results by combining a lagrangian relaxation based heuristic with branch and bound. Bienstock and Günlük (1996) and Magnanti et al. (1995) provide an insight into the use of cutting plane algorithms for solving multi-commodity capacitated flow problems. Recently Crainic et al. (2001) have shown the effectiveness of a combined approach based on the Lagrangian relaxation method and bundle methods. They suggest that such judicious combinations are desirable for adaptation into a parallel computing environment.

The Benders decomposition method is based on the idea of decomposing the problem in two subproblems namely a master problem and a subproblem, in which the solution of the latter is utilized to generate a cut for the master problem which is repeated until some predetermined termination condition is met. Benders decomposition has been successfully applied to many of these problems, and we refer the reader to a review by Costa (2005). Finally, since we are considering large NP Hard problems, certainly heuristics are useful to solve comparatively large instances of these problems (Gendron et al., 1998)

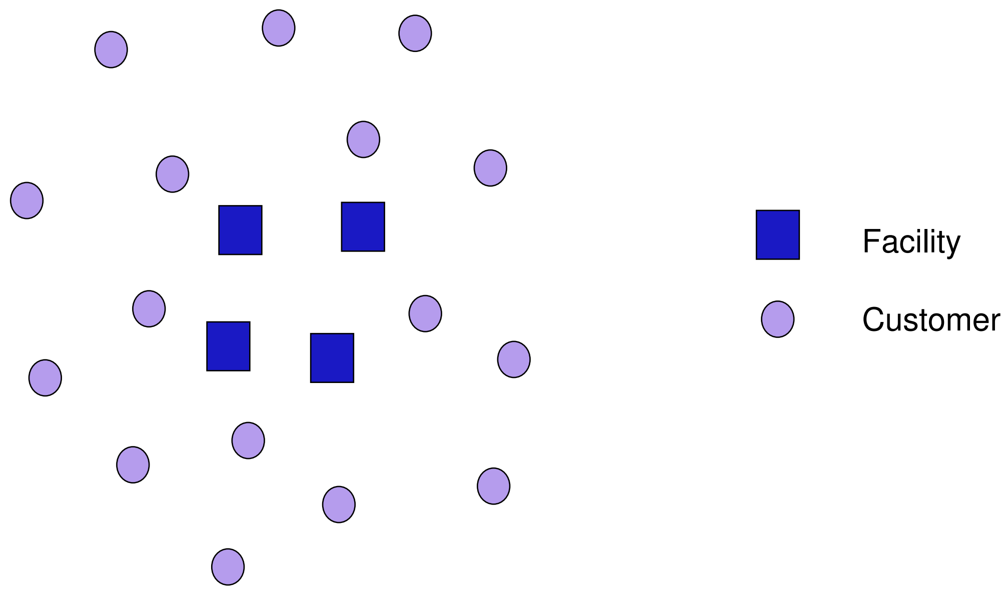
Similarly to the network design problem, the commodity flows in our problem are also routed via capacitated arcs. However, the problem at hand distinguishes itself from the general network design problems in the sense that it considers the consolidation and deconsolidation activities explicitly. In addition, the transportation costs on the transfer links from the consolidation centers to deconsolidation centers is not linear, but a step function of the quantity being transferred. In other words, these transfer links can have multiple FTLs (Full Truck Loads) installed. Further, as opposed to network design problems, our problem has a structure that requires use of, at most, three arcs or direct shipments which is not the case in general network design problems.

II.3. Facility Location

Industry and government are both faced with location decisions such as how many facilities to create, where to locate them, how large they should be, and what customers to assign to which facility? Location decisions have an impact on the service quality and the cost of service, which are two most important decisions for any industrial or governmental entity. Some typical applications of location problems are warehouse location, plant location, hospital /school/fire station location problems. Depending upon the application, there are a number of traditionally studied problems. We refer the reader to the well-known text books on the subject by Francis et al. (1992), Love et al. (1988) and Daskin (1995).

A rich literature exists on a number of location problems, both continuous and discrete. However, we will review only discrete location problems as they are relevant to our problems. In all of the discrete facility location problems, the set of possible facility locations on a network and the set of customers are given as shown in Figure

Figure 7 Facility Location



7. Usually there are fixed costs associated with opening a facility and variable costs associated with the use of the facility. Customer demands are to be satisfied from the selected facilities. There are transportation costs associated with the transportation of goods from facilities to the customers.

The simplest facility location problem is to select a set of locations from a set of potential locations to set up facilities such that every client's demand is satisfied by one or more facilities. The problem is called as uncapacitated or capacitated problem depending on whether facilities are assumed to have unlimited or limited capacities. Below we review model formulation for these two cases. For notation, let $\mathcal{I} = \{1, \ldots, n\}$ be the set of potential facility locations, and $\mathcal{J} = \{1, \ldots, m\}$ be the set of clients.

II.3.1. Uncapacitated Facility Location Problem

In uncapacitated facility location problem (UFLP) there is no restriction on the number of clients a facility can serve. The following is a mixed integer programming (MIP) formulation (Daskin, 1995).

Parameters:

c_j cost of placing a facility at location j.

h_{ij} total cost of satisfying demand of client i from a facility at j.

Decision Variables:

x_j 1 if a facility is placed at location j, 0 otherwise.

y_{ij} fraction of the demand of client i that is satisfied by facility at j

Objective and Constraints:

$$\min \sum_{j \in \mathcal{J}} c_j x_j \quad + \quad \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} h_{ij} y_{ij} \tag{2.17}$$

subject to

$$\sum_{j \in \mathcal{J}} y_{ij} = 1 \qquad \forall i \in \mathcal{I} \tag{2.18}$$

$$y_{ij} \leq x_j \qquad \forall i \in \mathcal{I}, j \in \mathcal{J} \tag{2.19}$$

$$x \in \{0,1\}^n, y \in \mathbb{R}_+^{mn} \tag{2.20}$$

The first term in the objective function (2.17) represents the fixed cost of locating the facilities, whereas the second term is total cost of serving the clients. Constraint (2.18) ensures that every client's demand is exactly met and constraint (2.19) ensures that a client i is not served from a location j if a facility is not placed in that location

($x_j = 0$).

The UFLP belongs to the class of NP-Hard problems. Several solution methods including heuristic, implicit enumeration, and dual based have been investigated for UFLP. UFLP is one of the most important classical location problems because it appears as a subproblem or special case of other problems. A complete review of the literature is beyond the scope of this work. Therefore, we refer the reader to Daskin (1995) and Cornuejols et al. (1990) for discussion of the various solution approaches for UFLP including ADD-DROP heuristic, Lagrangian and Dual based approach (Erlenkotter, 1978).

II.3.2. Capacitated Facility Location Problem

A natural extension of the UFLP is obtained by considering capacity restrictions on the facilities. The *capacitated facility location problem* (CFLP) involved selecting a set of locations from a set of potential locations in order to set up facilities that have a fixed capacity depending on where they are located. Every client has a fixed demand that is satisfied by one or more facilities. The objective is to minimize facility placement costs and client service costs, subject to location and capacity constraints. The following is a mixed integer programming (MIP) formulation (Daskin, 1995).

Parameters:

u_j capacity of a facility located at j.

b_i total demand of client i.

c_j cost of placing a facility at location j.

h_{ij} total cost of satisfying demand of client i from a facility at j.

Decision Variables:

x_j 1 if a facility is placed at location j, 0 otherwise.

y_{ij} fraction of the demand of client i that is satisfied by facility at j

Objective and Constraints:

$$\min \sum_{j \in \mathcal{J}} c_j x_j \quad + \quad \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} h_{ij} y_{ij} \tag{2.21}$$

subject to

$$\sum_{j \in \mathcal{J}} y_{ij} = 1 \qquad \forall i \in \mathcal{I} \tag{2.22}$$

$$\sum_{i \in \mathcal{I}} b_i y_{ij} \leq u_j x_j \qquad \forall j \in \mathcal{J} \tag{2.23}$$

$$\sum_{j \in \mathcal{J}} u_j x_j \geq \sum_{i \in \mathcal{I}} b_i \tag{2.24}$$

$$y_{ij} \leq x_j \qquad \forall i \in \mathcal{I}, j \in \mathcal{I} \tag{2.25}$$

$$x \in \{0,1\}^n, y \in \mathbb{R}_+^{mn} \tag{2.26}$$

In the formulation above, the first term in the objective function (2.21) represents the fixed cost of locating the facilities whereas the second term is the total cost of serving the clients. Constraint (2.22) ensures that every client's demand is exactly met. Constraint (2.23) ensures that no client is served from a location j if a facility is not placed in that location ($x_j = 0$), and total service is only up to its capacity if a facility is placed. Constraint (2.24) specifies that the total capacity should be at least equal to total demand. It is called a surrogate constraint since it can be obtained by summing constraint (2.23) over all facilities. Constraints (2.25) are disaggregation constraints; although redundant in the formulation, they may tighten certain relaxations.

CFLP belongs to the class of NP-Hard problems and is a well studied classical facility location problem. Several methods including heuristics, Lagrangian relaxation (Agar and Salhi, 1998), Branch and Bound (Akinc and Khumawala, 1977), Benders decomposition (Davis and Ray, 1969; Wentges, 1996) and a combination of Benders and Lagrangian based cross decomposition (Roy, 1986) have been suggested for the solution of CFLP. Reviewing the complete literature is beyond the scope of this dissertation; however we refer the reader to reviews by Beasley (1993), Magnanti and Wong (1986) and Sridharan (1995).

CFLP can be extended in various ways to capture different business practices and we present below the literature related to two extensions which are are relevant to our problem. The first extension is known as Single Source Capacitated Facility Location (SSCFLP) in which a customer is restricted to assignment to exactly one facility. In another extension known as CFLP with modular capacity, for each potential facility location, there is a finite and discrete set (modules) of allowable capacities, and the objective is to choose the subset of facilities that satisfy the demand at minimum cost. We will show later that our problem ONDP is related to SSCFLP and that TNDP is related to SSCFLP with modular capacities. CFLP, where customers are allowed to be serviced by multiple facilities, is related to our problem SNDP.

The SSCFLP belongs to the class of NP-Hard problems. This problem has gotten the attention of the research community because of its applicability to a number of practical scenarios and because of the algorithmic challenges involved. For the SSCFLP, three solution approaches have been explored in the literature: Lagrangian relaxation (LR), exact methods, and heuristics. The Lagrangian relaxation approaches differ from each other in terms of the sets of constraints that are relaxed, the associated subproblem solution methods and methods of obtaining upper bounds. We refer to Barcelo and Casanovas (1984); Hindi and Pienkosz (1999); Holmberg et al. (1999); Pirkul (1987) and Sridharan (1993) for LR approaches that relax the assignment constraints, to (Klincewicz and Luss, 1986) for a LR approach that relaxes the capacity constraint, and to (Agar and Salhi, 1998; Beasley, 1993) for LR approaches that relax both the capacity and assignment constraints. Recently, Cortinhal and Captivo (2003) suggest the use of tabu search in an LR framework to obtain feasible solutions as upper bounds. However, obtaining and keeping the feasibility in upper bounds calculations still appears to be a challenging issue. As reported in the above literature summary, relatively small instances have been solved effectively. The number of attempts to solve the SSCFLP using exact methods have been few which is not surprising as the problem belongs to the class of NP-hard problems. Neebe and Rao (1983) model the SSCFLP as a set partitioning problem and develop a column generation based branch-and-bound method. Holmberg et al. (1999) develop a repeated matching and Lagrangian based branch-and-bound algorithm. Diaz and Fernandez (2002) suggest a branch-and-price framework. Finally, in terms of heuristic approaches, Ahuja et al. (2004) develop a very large scale neighborhood (VLSN) search algorithm which is used in a multi-start framework with the initial solutions generated by the LR approach given by Holmberg et al. (1999). Delmaire et al. (1999) suggest a reactive greedy randomized adaptive search procedure coupled with a tabu search approach. The computational studies reported in these studies span relatively small sized problems, and particularly, as is generally the case in location problems, consider instances in which the number of potential locations is significantly fewer than the number of customers.

II.4. Service Network Design

Service network design is important for the success of express shipment businesses. Express shipment carriers must operate on a distribution network that enables them to satisfy customer demand with pick-up and delivery times within tight time windows at a minimum cost. Various forms of express shipment service network design problem have been studied. Kuby and Gray (1993) develop a capacitated, single-hub formulation for an express shipment problem for FedEx. Grünert and Sebastian (2000) identify activities in postal and express shipment planning operations and define corresponding optimization problems. Barnhart and Schneur (1996) present a model for express shipment and suggest a column generation based solution approach. Barnhart et al. (2002) develop a modelling framework for an express shipment delivery network design problem. They decompose the problem into a route generation problem that is solved using a branch-and-price-and-cut approach and a shipment movement problem that is solved with a branch-and-price approach and then suggest an iterative approach to solve the overall problem. Armacost et al. (2002) develop a composite variable formulation that provides stronger bounds, along with the flexibility to handle operational constraints, that make conventional formulation intractable. Based on this approach, in Armacost et al. (2004), they develop a system to solve an express shipment delivery network problem for UPS.

Load planning in service network design consists of determining how to route small shipments over a network to minimize the transportation cost while maintaining service levels (Powell and Sheffi, 1983). Powell (1986) formulates the LTL network design problem as a fixed-charge network design problem where the levels of service constraints are represented heuristically. He suggests a local improvement heuristic for problems involving over 300 terminals. Powell and Sheffi (1989) combine and extend these studies to develop an interactive system for network design in the motor carrier industry. Lamar and Sheffi (1987) develop a lower bound for this problem and suggest a solution scheme using a link-inclusion heuristic in an implicit enumeration framework. Lapierre et al. (2004) provide a model that addresses the case where there is at most one center on the route of a commodity from its origin to its destination, and they suggest tabu and variable neighborhood search approaches. We note that these authors consider only a single stop for each commodity whereas in our study we consider consolidation and associated truck capacity issues explicitly.

Leung et al. (1990) present a problem motivated by point-to-point delivery applications similar to ours. They present a mixed-integer non-linear formulation and a solution heuristic that decomposes the problem into two smaller subproblems. The first subproblem assigns the origin-destination pairs to the first and last centers on their route so that the implied flows between the center pairs are determined given these assignments and flow requirements, the second subproblem determines the minimum cost routing for center pairs. This is achieved in such a way that, for each center pair, the flow follows a single path but may visit multiple other centers subject to processing costs, and the flow may be split to different trucks on each of the paths it follows. Problems considered in this dissertation are motivated by exploiting economies-of-scale via consolidation, at the same time, avoiding delays and processing costs once the consolidation are formed. Therefore, at all three levels, we consider the consolidation of commodities into truckloads (truckload-trip in TNDP and SNDP), once formed, follows the shortest path, or some other preferred path, on its transfer between centers without further processing. In comparison to their decomposition based solution approach, our approach is different. In our case, we define the problem in three different levels of decision making; operational, tactical and strategic levels, and consider only the decision variables that are relevant to the particular planning horizon. Defining problems based on planning horizon allows us to exploit the problem structure and develop dedicated models and solution methods.

Relatively recent reviews in these areas appear in Crainic (1999, 2000) and in Campbell (2005).

II.5. Summary and Conclusions

In this chapter we presented a review of literature relevant to the problems addressed in this dissertation. The relevant literature can be classified in four categories, hub location, network design, facility location and service network design. Since our problem concerns the design of a distribution network and involves consolidation and deconsolidation activities, the hub location literature is naturally relevant to us, and since we are interested in routing commodities over a network, the general area of network design is also relevant. Furthermore, since each linehaul link can also be considered as a facility, and we consider center location decisions in TNDP, our problems are also related to facility location problems. Finally, due to our specific application area, studies in logistics service network design and load planning are also within our scope.

Hub location problems are related to our problem because of operational similarity as both are motivated by scale-economies derived from consolidation of loads. In this chapter, we presented a problem description and a representative mathematical model. We reviewed some of the solution methods and discussed the distinguishing features of our problems that preclude application of the hub location models and solution methods.

Since our problems involve routing the commodity flow on a network, we also presented literature review related to multicommodity network design problem.We gave a formal description of the problem followed by arc-based and path-based formulations. We discussed various solution methods for capacitated and uncapacitated versions of the problems. Since our problem models consolidation related activities explicitly, and has step function type cost for using transfer links, network design models are not suitable for our problem.

Since, in SNDP we address location decision related to consolidation and deconsolidation centers, and in ONDP and TNDP, the line-haul link can be considered as a facility, facility location literature is relevant to us. We presented models for uncapacitated and capacitated facility location problems and discussed extensions of CFLP, SSCFLP and SSCFLP with staircase capacity, that are related to our problem.

Furthermore, due to the specific application area, studies in logistics service network design and load planning are also within our scope. We reviewed several papers that focus on various applications in that area. Finally, since each transfer link can be considered to be a facility and the commodity as a customer, therefore we presented problem description, uncapacitated, capacitated and single source capacitated facility location problems formulations and their solution algorithms.

In the light of the discussion of related literature, we observe that the traditional models of LTL and intermodal transportation use hub-and-spoke and network design type approaches which lack one or more of the following:

1. commodity based routing decisions.

2. explicit consideration of economies of consolidation.

3. a special network structure with consolidation and deconsolidation centers.

4. single sourcing constraints that forces the commodity to flow on a single path in order to avoid unnecessary operational complexities and delays.

In the following chapters, we will present dedicated models and solution algorithms for the problems described in the previous chapter.

CHAPTER III

OPERATIONAL NETWORK DESIGN PROBLEM

In ONDP, we are given a network with multi-commodity flows where each commodity is defined by its unique pair of origin and destination nodes and a known required flow amount. The system is operated in such a way that the commodities are collected and consolidated into truckloads at consolidation centers, a linehaul transfer takes place for the consolidated loads, which are deconsolidated at deconsolidation centers and from there, the commodities are shipped to their final destinations. In addition to the network and commodity flows, we are also given a fleet of trucks . Additionally, we allow direct shipments between origin and destination nodes since this is preferred when the origin and destination nodes of a commodity are relatively close, and, thus, consolidation does not make economical sense. The decisions to be made in ONDP include

1. the assignment of trucks to linehaul transfer links

2. the assignment of commodities to a truckload shipment established on transfer links

3. the identification of commodities that are to be shipped directly

Such multi-commodity distribution network design problems appear frequently in various logistical applications where there is a significant flow of entities such as raw material, work-in-process, finished products, parcels/packages, information or passengers.

Under these operational and configurational characteristics, our purpose is to determine the linehaul links between the regional centers (consolidation and deconsolidation centers), the number of trucks assigned to each linehaul link, and the routing and consolidation of commodities so that the total cost of transportation is minimized. The total cost has four components: the costs of collection, distribution, linehaul transfer from consolidation to deconsolidation centers, and direct shipments.

The remainder of this chapter is organized as follows. Next, in Section III.1, we develop a binary integer program formulation for our problem. In Section III.2, we describe the basic ingredients of the heuristic solution approaches in detail, including our compound neighborhood functions. In Section III.3, we present three algorithmic frameworks including local search, simulated annealing, and tabu search, where, in each approach, we employ a solution neighborhood exploration strategy that relies on branching based on solution representation characteristics. In Section III.4, we present the results of our computational tests regarding the performance of the approaches and, in Section III.5, we provide a summary of our conclusions and future research directions.

III.1. The Model

In order to develop a mathematical formulation of our distribution network design problem by utilizing the auxiliary graph as described on page 8, we define the following additional notation.

Parameters:

w_i the amount of flow for commodity p_i

U capacity per full TL for long-haul transfers

β full TL transportation cost per mile between $\mathcal{J}$ and $\mathcal{K}$

α^f LTL transportation cost per unit per mile between $\mathcal{F}$ and $\mathcal{J}$

α^t LTL transportation cost per unit per mile between $\mathcal{K}$ and $\mathcal{T}$

α^{ft} LTL transportation cost per unit per mile between $\mathcal{F}$ and $\mathcal{T}$

d^f_{ij} distance between f_{p_i} and consolidation center j

d^t_{ki} distance between deconsolidation center k and t_{p_i}

d_{jk} distance between centers j and k

d^{ft}_i direct shipment distance for commodity p_i

$\mathcal{L}$ set of linehaul trucks (TL shipments), $l = 1, \ldots, |\mathcal{L}|$

Decision Variables:

z_{ijkl} 1 if commodity p_i is assigned to TL $l \in \mathcal{L}$ installed on linehaul link (j,k), 0 o.w.

y_{jkl} 1 if TL shipment l is assigned to the linehaul link (j,k), 0 o.w.

s_i 1 if commodity p_i is shipped directly from its origin to its destination , 0 o.w.

Then, the problem can be formulated as follows:

Objective and Constraints:

$$\text{Min} \quad \sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{L}} w_i\,(\alpha^f\,d^f_{ij} + \alpha^t\,d^t_{ki})\,z_{ijkl} + \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{L}} \beta\,d_{jk}\,y_{jkl} + \sum_{i\in\mathcal{P}} \alpha^{ft}\,w_i\,d^{ft}_i\,s_i \tag{3.1}$$

subject to

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{L}} z_{ijkl} + s_i = 1 \qquad \forall\, i \in \mathcal{P} \tag{3.2}$$

$$\sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} w_i\, z_{ijkl} \leq U \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} y_{jkl} \qquad \forall\, l \in \mathcal{L} \tag{3.3}$$

$$z_{ijkl} \leq y_{jkl} \qquad \forall\, i \in \mathcal{P}, j \in \mathcal{J}, k \in \mathcal{K}, l \in \mathcal{L} \tag{3.4}$$

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} y_{jkl} \leq 1 \qquad \forall\, l \in \mathcal{L} \tag{3.5}$$

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} y_{jk(l+1)} \leq \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} y_{jkl}, \qquad l = 1, \ldots, |\mathcal{L}| - 1 \tag{3.6}$$

$$z_{ijkl}, s_i, y_{jkl} \in \{0, 1\} \qquad \forall\, i \in \mathcal{P}, j \in \mathcal{J}, k \in \mathcal{K}, l \in \mathcal{L} \tag{3.7}$$

In the objective function given by expression (3.1), the first term represents the total transportation cost for collection and distribution operations in graphs $G^C(\mathcal{F} \cup \mathcal{J}, A_{FJ})$ and $G^D(\mathcal{K} \cup \mathcal{T}, A_{KT})$, respectively; the second term represents the total transportation cost for linehaul transfers using TL shipment in $G^L(\mathcal{J} \cup \mathcal{K}, A_{JK})$; and the third term represents the total transportation cost for commodities shipped directly from their origins to their destinations. Constraint set (3.2) states that each commodity is either included in a TL shipment or shipped directly. Constraint set (3.3) ensures that the total weight of the commodities assigned to a TL shipment does not exceed the capacity of a truck. We estimate the maximum size of the set $\mathcal{L}$ as $\lceil \sum_{i\in\mathcal{P}} w_i/(0.75\ U)\rceil$ which is obtained by assuming that, on average, 75% full TLs justify consolidated shipments. While this estimate is conservative, in our numerical studies, we observe that the set $\mathcal{L}$ is never exhausted. Constraint set (3.4) ensures that a commodity can be assigned to a TL shipment on a particular linehaul transfer link only if this link has that TL installed on it. This constraint, although redundant since it is implied by the previous constraint, helps greatly in obtaining better lower bounds when linear programming relaxation is used in a branch-and-cut framework. Constraint set (3.5) ensures that each potential TL is assigned to, at most, one linehaul transfer link. Constraint set (3.6) helps to reduce the symmetry in allocating linehaul trucks to transfer links and constraint set (3.7) imposes standard binary restrictions on the decision variables.

Notice the simple, yet effective consideration of transportation *economies-of-scale* realized through TL consolidations using capacitated trucks. In addition, any fixed costs associated with TL shipments can also be easily incorporated into the second term of the objective function.

For the linehaul TL shipments between the regional centers, we assume that the travel follows the shortest path on the physical network or that it is specified according to possible shipment routes. For the latter case, which can provide more realistic estimates of center-to-center (for consolidated linehaul) routes, one needs to observe several operational considerations faced by the TL transportation industry. These issues include, among others, driver turnover rates, empty dispatch mileage, and additional circuitry caused by splitting direct routes into segments. A mathematical model that incorporates these issues and an efficient solution procedure that addresses network design for multi-zone dispatching in the TL industry is given in Üster and Maheshwari (2007).

Our problem is also related to the single-source capacitated facility location problem (SSCFLP) where the latter is a special case of our problem. In SSCFLP, given the potential facility locations with known capacities, the objective is to minimize the total cost of location and transportation while satisfying customer demands in such a way that each customer is assigned to a single facility. A solution to our problem consists of some commodities being assigned to TL shipments on linehaul transfer links and the remaining commodities, if present, being shipped via direct shipments. One can view a potential TL shipment on a transfer link as a capacitated facility and the commodities as the customers. In order to model the direct shipments, we can create a pair of dummy consolidation-deconsolidation centers for each commodity and define the cost of assigning the commodity to its dummy center pair as equivalent to the cost of direct shipment with zero collection and distribution costs (costs associated with other commodities and the dummy link are simply taken as infinity). Then, clearly, our problem generalizes SSCFLP. However, from the perspective of a typical SSCFLP setting, our problem is too large for state-of-the-art methods to solve. For example, let M be the number of centers and L be the number of potential TL shipments. Recall that N is the number of commodities. Then, an equivalent SSCFLP will have $L\,M^2 + N$ potential facilities and N customers. In an instance with 500 commodities on a network with $M = 6$ and $L = 20$, the equivalent SSCFLP will have $20 \times 36 + 500 = 1220$ potential locations and 500 customers. Typically, SSCFLPs that are much smaller in size are considered difficult in the literature Cortinhal and Captivo (2003). Furthermore, an equivalent SSCFLP version of our problem has a significantly higher number of potential locations than the number of customers which is quite the opposite of a typical facility location problem where the number of potential locations is very small when compared with the number of customers. Therefore, even a small instance of our problem is actually prohibitively large for state-of-the-art methods to solve efficiently.

Büdenbender et al. (2000) consider a related problem, the direct flight network design problem, that arises in the area of mail transportation. Given the freight requirements between origins and destinations, where a pair is a commodity, the objective is to determine the routing of commodities through airports (centers) in such a way that the total transportation cost is minimized. Again, direct shipments are not allowed. The authors present a hybrid tabu search/branch-and-bound algorithm. As opposed our problem, they assume that the flow requirement of a commodity can be split on different routes during transportation. Thus, the resulting model is posed as a capacitated facility location problem with multi-sourcing that involves side constraints to address limitations in the number of flights between airport pairs and the number of possible take-offs and landings at the airports. Although we do not consider similar side constraints, again, they can be incorporated into our model and solution approaches via preprocessing and feasibility checks on the neighborhood solutions.

We use the formulation (3.1)-(3.7) to find optimal solutions to small problem instances by using CPLEX 9.0 which employs branch-and-cut methodology with several cut options, preprocessing and upper bound heuristics. For relatively smaller instances, the formulation can be solved to optimality efficiently; however, the computational time and memory requirements become quite prohibitive for large problem instances. Since our interest is in solving large instances, in this chapter we provide three heuristic solution approaches that exploit the problem structure to efficiently obtain good solutions.

III.2. Ingredients of the Heuristic Algorithms

In this section, we first describe the solution representation and objective function evaluation method which is frequently used during the heuristic solution process. Later, we provide two alternative construction heuristic approaches which are employed to generate initial feasible solutions as inputs to the solution improvement (heuristic search) algorithms. Also in this section, we present the details of our compound neighborhood functions which are important ingredients of the three heuristic search algorithms to be described later.

III.2.1. Solution Representation and Objective Function Evaluation

We observe that, in any feasible solution to our problem, the set of commodities is partitioned into disjoint and mutually exclusive subsets. One of these subsets includes commodities shipped directly from their origins to their destinations. We denote this subset by $\mathcal{D}$. On the other hand, each of the other subsets corresponds to commodities that are included in the same TL shipment l, i.e., these commodities are consolidated together into the same TL shipment. To represent each subset of commodities forming a TL shipment, we use the notation C_l. Also, we denote the set of all such subsets by $\mathcal{C}$. Note that, for feasibility, we ensure that the total weight of the commodities in a TL shipment does not exceed the truck capacity. Then, the pair $(\mathcal{C}, \mathcal{D})$ represents a feasible solution to our problem. Henceforth, for notational simplicity, we represent a solution $(\mathcal{C}, \mathcal{D})$ by $\mathcal{S}$.

In order to calculate the goodness of a given solution $\mathcal{S}$, we need to calculate the total transportation cost it implies. For each TL shipment l, we need to determine the transfer link (j, k), where $j \in \mathcal{J}$ and $k \in \mathcal{K}$, that the commodities in C_l use. For each C_l, we simply pick the link (j, k) that provides the lowest collection-transfer-distribution cost (first two terms in the objective function) via total enumeration over the $|\mathcal{J}| \times |\mathcal{K}|$ transfer links. On the other hand, the cost associated with the commodities in $\mathcal{D}$ can be calculated directly, providing the value of the last term in the objective function. Then, the objective function value associated with $\mathcal{S}$ is the sum of the costs associated with the commodity subsets it involves.

III.2.2. Construction Heuristics

Based on our solution representation described above, the construction of an initial feasible solution clearly consists of partitioning commodities into (i) TL shipments and (ii) the set of commodities shipped directly. There are a number of possible ways to obtain such partitions. Here, we describe two initial solution construction methods that are partially greedy in nature. Both methods incorporate randomness in order to support a multi-start framework for improvement heuristics. We also note that in both of these methods we generate purely TL shipments, since the direct shipment criterion for commodities is highly subjective at this stage. Later, in the improvement heuristics, the direct shipment set $\mathcal{D}$ is populated and modified during the search procedure.

In the first construction method, we randomly pick commodities to form the TL shipments, one TL shipment at a time, without exceeding truck capacity. The procedure stops when all the commodities are assigned. Then, the transfer link selection for each TL shipment and the associated cost evaluation is performed in a greedy fashion as described above. This procedure, which we refer to as C-RC() (since it **R**andomly picks **C**ommodities), is given in Display 1.

Display 1 Construction Procedure C-RC()

Initialize: $\mathcal{D} = \emptyset$, $\mathcal{C} = \emptyset$;
while $\mathcal{P} \neq \emptyset$ **do**
    $C_{\text{temp}} = \emptyset$, $v = 0$
    while $v \leq U$ **do**
        Randomly pick a commodity $i \in \mathcal{P}$
        if $(w_i \leq U - v)$ **then**
            $C_{\text{temp}} = C_{\text{temp}} \cup \{i\}$
            $v = v + w_i$
            $\mathcal{P} = \mathcal{P} \setminus \{i\}$
        end if
    end while
    $\mathcal{C} = \mathcal{C} \cup C_{\text{temp}}$
end while
Return $\mathcal{S} = (\mathcal{C}, \mathcal{D})$

In the second method, we randomly select linehaul transfer links, one link (j, k) at a time, and assign commodities to a TL shipment installed on the selected link. After selecting a transfer link (j, k), we assign commodities in a greedy fashion based on their proximity, as measured by the associated collection and distribution costs, to the link. Random transfer link selection is repeated until each commodity is assigned to a TL shipment. We allow a link (j, k) to be selected more than once since a transfer link can have multiple TL shipments on it. Display 2 includes the details of this procedure, which we call C-RL() (since it **R**andomly picks **L**inks).

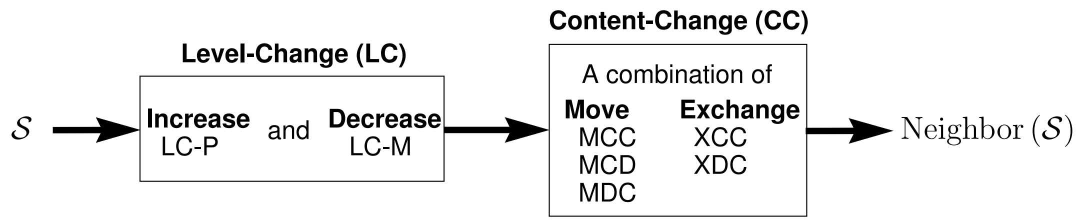
Display 2 Construction Procedure C-RL()

Initialize: $\mathcal{D} = \emptyset$, $\mathcal{C} = \emptyset$
while $\mathcal{P} \neq \emptyset$ **do**
    Randomly pick a link (j, k), $j \in \mathcal{J}$, $k \in \mathcal{K}$
    $C_{\text{temp}} = \emptyset$
    Calculate $\gamma_i = w_i\,(\alpha^f\, d^f_{ij} + \alpha^t\, d^t_{ik})$, $\forall\, i \in \mathcal{P}$
    Sort commodities in their increasing γ_i values
    Include first κ commodities such that $\sum_{i=1}^{\kappa} w_i \leq U$ into C_{temp}
    $\mathcal{P} = \mathcal{P} \setminus C_{\text{temp}}$
    $\mathcal{C} = \mathcal{C} \cup C_{\text{temp}}$
end while
Return $\mathcal{S} = (\mathcal{C}, \mathcal{D})$

III.2.3. Components of Compound Neighborhoods

In any given solution $\mathcal{S}$, we identify three key attributes including the consolidation level defined as the number of TL shipments $|\mathcal{C}|$, the composition of the set $\mathcal{D}$, and the composition of TL shipments $C_l \in \mathcal{C}$. Given a solution, a neighborhood function modifies these key attributes in order to generate neighboring solutions in a heuristic search framework. Since the neighborhood functions that we develop for this purpose utilize simple operations in various combinations for modifying the key attributes, we call them compound neighborhood functions. There are two essential components of a compound neighborhood function: the *Level-Change* (LC) and the *Content-Change* (CC). The LC component perturbs the consolidation level $|\mathcal{C}|$ in a solution $\mathcal{S}$, and the CC component modifies the contents of TL shipments $C_l \in \mathcal{C}$ and the direct shipment set $\mathcal{D}$. We define a compound neighbor of a given solution $\mathcal{S}$ as a solution obtained by first applying an operation in the LC component followed by a combination of operations in the CC component. In the latter, a specific combination is called a CC method. These components and their operations are outlined in the Figure 8.

Figure 8 Components of Compound Neighborhoods



The **LC component** is comprised of two operations which are abbreviated as LC-P() and LC-M(). Given a solution $\mathcal{S}$, LC-P($\mathcal{S}$) gives a new solution with a consolidation level $|\mathcal{C}|+1$ whereas LC-M($\mathcal{S}$) gives a new solution with a consolidation level $|\mathcal{C}|-1$. In operation LC-P(), the consolidation level is increased by one TL shipment by bringing commodities from $\mathcal{D}$, or from other consolidated shipments, into $\mathcal{C}$ via consolidating them to form a new TL shipment. While trying to increase the number of TL shipments $|\mathcal{C}|$, one of the following cases can occur:

Case 1: If $\mathcal{D}$ is empty, then, to form a new consolidated shipment, we pick a pair of commodities from two separate TL consolidations where each has at least three commodities. We randomly select commodities from their current consolidations and form a new TL shipment with these two commodities and update $\mathcal{S}$.

Case 2: If the total demand of the commodities in set $\mathcal{D}$ is less than the truck capacity U, then we send all the commodities in $\mathcal{D}$ by a consolidated TL shipment, set $\mathcal{D}$ as an empty set and update $\mathcal{S}$.

Case 3: If total demand of commodities shipped via direct shipments currently is greater than U, we pick commodities randomly from $\mathcal{D}$ in order to form a new TL shipment C without exceeding the truck capacity. We revise set $\mathcal{D}$ accordingly and add the newly formed C into set $\mathcal{C}$.

In operation LC-M(), we reduce the consolidation level of a solution by one TL by disaggregating one of the TL shipments and sending its content via direct shipments. We disaggregate the TL shipment for which the increase in the objective function value of the resulting solution is minimum.

The **CC component** modifies the composition of sets $\mathcal{D}$ and $C_l \in \mathcal{C}$ using local search with five simple neighborhood functions listed in two groups in Figure 8. The first group, Move, includes three move operations which correspond to i) moving a commodity from the set $\mathcal{D}$ to a set C_i (MDC), ii) moving a commodity from some C_i to set $\mathcal{D}$ (MCD), and iii) moving a commodity from a set C_i to a set C_j (MCC) where $i \neq j$. The second group, Exchange, includes two pair-exchange operations which correspond to i) exchanging a pair of commodities between the set $\mathcal{D}$ and a set C_i (XCD) and ii) exchanging a pair of commodities between a set C_i and another set C_j (XCC) where $i \neq j$. These five simple neighborhood functions can be combined in several ways to prescribe different CC methods that generate a feasible (with respect to capacity constraints) neighboring solution of a given solution provided by the LC operation. Four such methods that, in turn, define four neighborhood functions can be classified under parallel type (CC-PN and CC-PLSN) and serial type (CC-SN and CC-SLSN) neighborhoods as described below.

Parallel Neighborhood (CC-PN): The parallel type method initially finds the first improving solution of a given solution using each of the five simple neighborhood functions separately (hence, the name parallel). Then, it picks the best one in terms of the objective function value as the neighboring solution.

Parallel Local Search Neighborhood (CC-PLSN): This method is similar to the previous one. However, we apply a complete local search routine starting with the given solution using each one of the five simple neighborhood functions separately. An iteration of each such complete search routine is terminated as soon as an improvement is obtained, and the complete routine is terminated when no improving solution is found. This way, we again obtain five separate solutions associated with the given solution, and we pick the best one as the neighboring solution.

Serial Neighborhood (CC-SN): This first serial type method utilizes the five simple neighborhoods as does the CC-PN, but in a sequential manner rather than a parallel manner. In particular, the first improving solution in a simple neighborhood provides the initial solution for the next simple neighborhood. Observe that there are several sequences of the five simple neighborhoods that can be employed. We differentiate two specific sequences: one follows the LC-P() operation and the other follows the LC-M() operation. In the former, we denote the associated CC method as CC-SN-P, and starting with a given solution, we use the sequence MDC→XCD→XCC→MCC→MCD→XCC. In the latter, we employ the sequence XDC→MDC→MCD→MCC→XCD, and we denote the corresponding CC method by CC-SN-M. The neighboring solution is given by the solution obtained after the final operation, i.e., after XCC when CC-SN-P is used and XCD when CC-SN-M is used.

Serial Local Search Neighborhood (CC-SLSN): Analogous to the relationship between the parallel type methods, this method is similar to the CC-SN. In the CC-SLSN, instead of finding the first improving solution, we conduct a complete local search with each simple neighborhood function in sequence. Similarly to the CC-SN method, we have CC-SLSN-P which follows a LC-P() operation and

CC-SLSN-M which follows a LC-M() operation. We employ the same sequences specified above for CC-SN-P and CC-SN-M in the CC methods CC-SLSN-P and CC-SLSN-M, respectively. For example, in the CC-SLSN-P, given a solution obtained by the LC-P() operation, we first apply a complete local search using the MDC neighborhood, and we accept the first improving solution at each iteration and continue the iterations until no improving solution is found. The solution thus obtained is then used as the initial solution for a complete local search employing the XCD neighborhood, and so on. The final solution obtained after the local search with the XCC neighborhood is taken as the neighboring solution of the starting solution which was provided by the operation LC-P().

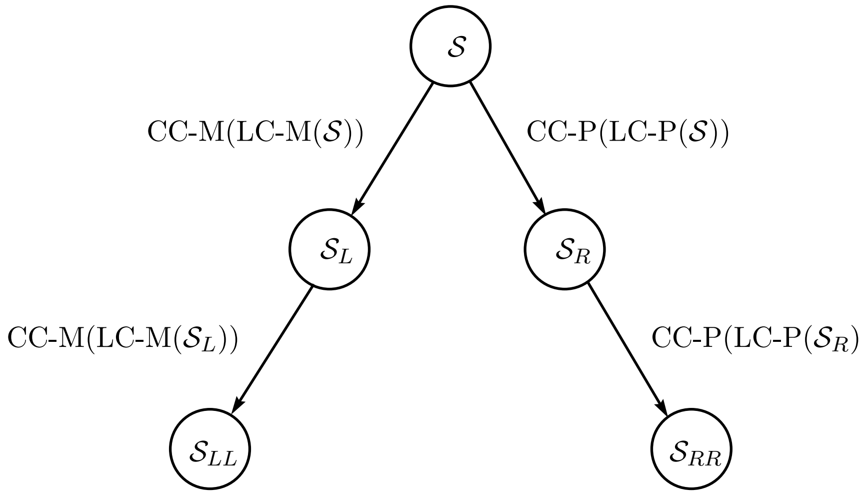
A few remarks regarding the above neighborhood functions are in order. First, for both of the serial type neighborhoods, there are several possible sequences. We determined the above mentioned sequences after careful analysis of many possibilities which explore the solution space more intensely after their respective level-change component specifics. Thus, these sequences are determined after extensive empirical testing. Second, the use of the LC and CC components in this fashion facilitates the incorporation of two desired characteristics of any heuristic search procedure. The LC component promotes *diversification* during the search of the feasible solution space. On the other hand, the methods of the CC component provide the opportunity for *intensification* in a solution subspace via the combined use of simple neighborhood functions.

III.2.4. Generic Notation and Branching

We define the generic notation that we use in describing the heuristic algorithms as follows. Without any specific reference, the function ConstructionHeuristic() refers to obtaining an initial solution which can be performed by applying either procedure C-RC() or C-RL(). For a CC component, we generically use CC-P() to refer to a solution obtained by ***local search*** with CC-PN, CC-PLSN, CC-SN-P, and CC-SLSN-P neighborhood functions. We also use CC-M() to represent a solution obtained by ***local search*** with CC-PN, CC-PLSN, CC-SN-M, and CC-SLSN-M neighborhood functions. In these local search procedures, we again take the first improving solution at each iteration. Recall that, although it is immaterial in parallel type content change methods, in serial type methods, the CC methods depend on the LC method used. Thus, we make the distinction with the suffix to indicate whether a specific CC method follows a LC-P() or LC-M() accordingly. For example, in a heuristic search procedure, LC-M($\mathcal{S}$) gives the solution obtained after the LC-M() operation is applied to a solution $\mathcal{S}$. If we employ a neighborhood function based on the PN method of content change, both CC-M($\mathcal{S}$) and CC-P($\mathcal{S}$) give the solution obtained after the local search with the CC-PN neighborhood function applied to $\mathcal{S}$. On the other hand, if we employ a neighborhood function based on the SN method, then CC-M($\mathcal{S}$) refers to the solution obtained after applying a local search with the CC-SN-M neighborhood to a solution $\mathcal{S}$.

We define a branching on a node representing a current solution $\mathcal{S}$ as shown in Figure 9. The left child of a node $\mathcal{S}$, denoted by $\mathcal{S}_L$, is a solution obtained by LC-M() followed by CC-M() and the right child of a node $\mathcal{S}$, denoted by $\mathcal{S}_R$, represents a solution obtained by LC-P() followed by CC-P(). Similarly, we obtain the solutions $\mathcal{S}_{LL}$ and $\mathcal{S}_{RR}$ as depicted in Figure 9.

Figure 9 Branching on a Solution $\mathcal{S}$ in ONDP



III.3. Heuristic Approaches

Our discussion in the previous section includes four distinct compound neighborhood functions which have the LC component with two operations as their common component and differ in terms of their CC component where we have four possible methods of combining simple neighborhood functions as outlined above. In this section, we describe three improvement heuristics, including local search, simulated annealing and tabu search procedures. Since each of the three heuristics can employ any one of the four compound neighborhood functions, we have a total of twelve solution procedures (excluding the options of construction heuristics).

III.3.1. Local Search with Deterministic Branching (LSDB)

In our local search with deterministic branching, given in Display 3, we start with an initial solution $\mathcal{S}^b$ obtained via a function ConstructionHeuristic(), which can be either C-RC() or C-RL(), and visit the solutions (nodes) $\mathcal{S}_R$, $\mathcal{S}_{RR}$, $\mathcal{S}_L$, and $\mathcal{S}_{LL}$ successively until an improving solution over the best solution, $\mathcal{S}^b$, is obtained. This search, called BranchSearch($\mathcal{S}$), is given in Display 4. As soon as an improving solution is found, we assign the corresponding solution as the new $\mathcal{S}^b$, designate it as the root node in our branching (Figure 9), and call the BranchSearch($\mathcal{S}^b$). This recursive process is continued until none of the four nodes of the branching associated with the current best solution provides an improvement. Since our construction heuristics are randomized, we also incorporate a multi-start aspect to LSDB as indicated in Display 3. The final solution $\mathcal{S}^f$ is simply the best solution obtained from all of the starts of the procedure.

Display 3 LSDB Algorithm

```
initialize  Z(S^f) = ∞, start=0, MAXSTART=15
while start < MAXSTART do
    S^c = ConstructionHeuristic(), S^b = S^c
    S^b = BranchSearch(S^b)
    if Z(S^b) < Z(S^f) then
        S^f = S^b
    end if
    start ++
end while
RETURN Z(S^f) and S^f
```

Display 4 BranchSearch($\mathcal{S}$) Function

if $Z(\mathcal{S}_R) < Z(\mathcal{S})$ **then**
    $\mathcal{S} = \mathcal{S}_R$
    BranchSearch($\mathcal{S}$)
else if $Z(\mathcal{S}_{RR}) < Z(\mathcal{S})$ **then**
    $\mathcal{S} = \mathcal{S}_{RR}$
    BranchSearch($\mathcal{S}$)
else if $Z(\mathcal{S}_L) < Z(\mathcal{S})$ **then**
    $\mathcal{S} = \mathcal{S}_L$
    BranchSearch($\mathcal{S}$)
else if $Z(\mathcal{S}_{LL}) < Z(\mathcal{S})$ **then**
    $\mathcal{S} = \mathcal{S}_{LL}$
    BranchSearch($\mathcal{S}$)
end if
RETURN $\mathcal{S}$

We note that, since tree based search explores only improving nodes, it may terminate at a local minimum. Furthermore, since the actions of perturbation components LC-P() and LC-M() are capable of cancelling the actions of one another if allowed to do so, the tree based search can potentially lead to cycling. However, a careful study of the procedures reveals that steps can be taken to prevent cycling. Note that if two solutions $(\mathcal{C}_1, \mathcal{D}_1)$ and $(\mathcal{C}_2, \mathcal{D}_2)$ have $|\mathcal{C}_1|$ same as $|\mathcal{C}_2|$, then they are each other's neighbors in the sense that it is possible to obtain one from the other by sequences of moves and pair-exchanges. On the other hand, if two solution have different numbers of consolidated truckloads then its impossible to obtain one from the other by sequences of moves and pair-exchanges. It implies that the solutions

$\mathcal{S}_L$, $\mathcal{S}_R$, $\mathcal{S}_{LL}$ and $\mathcal{S}_{RR}$ can never be same as the solution at node $\mathcal{S}$. Since procedures LC-P() and LC-M() are capable of producing solution with modified consolidation levels and they cancel the effect of one another, the only possibility of cycling arises when the solution $\mathcal{S}_{LR}$ or $\mathcal{S}_{RL}$ can possibly be same as the solution $\mathcal{S}$.

One of the ways to avoid cycling of the solution $\mathcal{S} = (\mathcal{C}, \mathcal{D})$ at a given node is to record the set $\mathcal{D}$ and make sure that set $\mathcal{D}'$ in the solution $\mathcal{S}_{RL}$ or $\mathcal{S}_{LR}$ is different than the original set $\mathcal{D}$. Comparison of two sets of commodities shipped directly is appropriate because it is computationally less expensive to check if $\mathcal{D}_1 \equiv \mathcal{D}_2$ than comparing the sets $\mathcal{C}_1$ and $\mathcal{C}_2$. We can also compare the objective function values of the two solutions $(\mathcal{C}_1, \mathcal{D}_1)$ and $(\mathcal{C}_2, \mathcal{D}_2)$ instead of comparing the sets $\mathcal{D}$ and $\mathcal{D}'$ as a check for cycling. Since, in our implementation we already have the objective function value for every new solution, we do not need to recompute objective functions values and hence we use the comparison of objective function values to detect the cycling instead of comparison of the $\mathcal{D}$ sets. Once we have detected cycling, we generate another neighborhood solution with procedure MCC and continue the tree search. Since, the tree search solution quality depends upon the initial solution, we implemented tree search with multiple initial solutions.

III.3.2. Simulated Annealing with Biased Branching (SABB)

The probabilistic acceptance of non-improving solutions (uphill moves) during a local search heuristic is the main feature of the simulated annealing (SA). Due to this property, a neighborhood search procedure employing SA may accept solutions that a typical local search procedure does not, and, thus, it provides the opportunity to reach a better local optimum. The SA framework is relatively easy to implement, and it has been an effective meta-heuristic for solving combinatorial optimization problems. For a detailed discussion of its features, we refer the reader to Aarts et al.

(1997); Sait and Youssef (1999).

In our simulated annealing algorithm, we again utilize the branching on the solutions obtained during the search. However, in this case, instead of making the selection in a deterministic way as in the LSDB, we suggest a probabilistic branching strategy that introduces bias into the search in such a way that the iterates tend to move to the region of the solution space that is more likely to include good solutions. Thus, we abbreviate this method as Simulated Annealing with Biased Branching (SABB). In general, the SABB algorithm is an iterative procedure where in each iteration, an inner loop, given by the Metropolis procedure, is performed. the Metropolis (loop) itself is an iterative procedure that relies on certain overall algorithmic parameters specified as the so-called cooling schedule. In Metropolis, we first generate a new solution ($\mathcal{S}^n$) using solution obtained via branching and evaluate its goodness (objective function value). This solution generation, given in lines 6-11 in Display 5, uses a biased branching strategy and it is particularly specific in our case as described in detail below. If $\mathcal{S}^n$ improves upon the best solution ($\mathcal{S}^b$) to date, we update the best and current ($\mathcal{S}^c$) solutions and start a new iteration of Metropolis. On the other hand, if $\mathcal{S}^n$ is non-improving, we accept it as the current solution with a probability $e^{-\Delta/\mathtt{T}}$ where Δ is the absolute difference between the current and new solution and $\mathtt{T}$ is an algorithm parameter known as temperature. This mechanism provides an opportunity for accepting the uphill moves mentioned above. The parameter $\mathtt{T}$ is usually high for initial Metropolis runs so the acceptance probabilities are high and diversification in the search is promoted. After each Metropolis run, the temperature is decreased before the next one starts, thus providing an overall decreasing sequence of temperatures, usually in a geometric fashion. This is achieved using a factor γ (typically a value less than and close to one), i.e., $\mathtt{T}$ is updated as $\gamma\,\mathtt{T}$. Each Metropolis procedure is executed at a fixed temperature for a certain number of iterations `M` which is another algorithm parameter. Similar to `T`, we also update the parameter `M` after each Metropolis run using another factor ϕ, i.e., `M` is updated with ϕ`M`; however, in this case, we choose a factor value that is greater than one. A cooling schedule set with these general characteristics promotes intensification in the search as the overall algorithm proceeds while encouraging diversification to reach regions with good solutions in the initial stages. The overall SABB Algorithm is terminated when the required number of iterations in a Metropolis loop exceeds a preset algorithm parameter value `MAX_M`. The complete SABB algorithm is given in Display 5. Note that the algorithm parameter values specified in the initialization step in Display 5 are obtained after some fine-tuning and used in our computational tests presented in Section III.4.

Of particular interest, in lines 6-11 of SABB procedure given in Display 5, we consider branching (see Figure 9) at each iteration of the Metropolis to generate a random solution. For this purpose, we incorporate a probabilistic node selection strategy. Specifically, a key feature of our compound neighborhood function is the branching which introduces two initial and two subsequent directions to modify the consolidation level $|\mathcal{C}|$ of a solution mainly using the operations LC-M() and LC-P(). The change in consolidation level can be seen as a neighborhood search direction. Knowing that the search direction plays an important role in determining the effectiveness of the simulated annealing procedure, we develop a branching strategy that dynamically determines search direction based on current consolidation level. In particular, we assume that, due to the economies-of-scale present in the cost structure, the optimal solution is more likely to have a consolidation level close to the consolidation level under perfect consolidation. Thus, the strategy exploits this problem structure to guide the search direction based on the consolidation level in the current solution. In order to describe this biased branching strategy, we first make some observations regarding solution characteristics. Let Q be the minimum number of trucks required to satisfy the shipment requirement under perfect consolidation with no direct shipments. The quantity Q can easily be calculated as $\lceil W(\mathcal{P})/u \rceil$ where $W(\mathcal{P}) = \sum_{i \in \mathcal{P}} w_i$. A solution $\mathcal{S}$ with no direct shipments, i.e., the case where the value of $|\mathcal{D}|$ is equal to zero, implies a value of $|\mathcal{C}|$ that is less than $N/2$, i.e., the maximum number of possible TL shipments, $N/2$. This corresponds to a solution in which $|\mathcal{D}|$ is equal to zero and exactly two commodities are sent in each of $N/2$ consolidated TL shipments. We can safely assume that a TL shipment does not include a single commodity since the direct shipment of this commodity would be more cost effective. On the other hand, a solution having no consolidations, i.e., $|\mathcal{C}|$ is equal to zero, implies that all of the N commodities are shipped directly, i.e., $|\mathcal{D}|$ is equal to N. In summary, $|\mathcal{D}|$ can be any number between 0 and N whereas $|\mathcal{C}|$ can vary anywhere between 0 and $N/2$. We observe that the variation in values of $|\mathcal{C}|$ between the range 0 to $N/2$ may facilitate exploration of the solution space via multiple search directions. Allowing a neighborhood search over all possible values of $|\mathcal{D}|$ and $|\mathcal{C}|$ is, however, practically infeasible. Thus, we concentrate on the value of $|\mathcal{C}|$ and its relation to Q.

Display 5 SABB Algorithm

initialize M=15, MAX_M=55,
T=500, $\gamma = 0.9$, $\phi = 1.2$
$\mathcal{S}^c$ = ConstructionHeuristic(), $\mathcal{S}^b = \mathcal{S}^c$
while M $\leq$ MAX_M **do**
  Set M$'$ = M
  repeat {*Metropolis Loop*}
    Calculate r = $|\mathcal{C}|/(|\mathcal{C}| + \lceil W(\mathcal{P})/u \rceil)$
    if $r >$ rand[0,1] **then**
      $\mathcal{S}^n = \arg\min\{Z(\mathcal{S}):\ \mathcal{S}^c_L, \mathcal{S}^c_{LL}\}$
    else
      $\mathcal{S}^n = \arg\min\{Z(\mathcal{S}):\ \mathcal{S}^c_R, \mathcal{S}^c_{RR}\}$
    end if
    $\triangle = Z(\mathcal{S}^n) - Z(\mathcal{S}^c)$
    if $\Delta < 0$ **then**
      $\mathcal{S}^c = \mathcal{S}^n$
      if $Z(\mathcal{S}^c) < Z(\mathcal{S}^b)$ **then**
        $\mathcal{S}^b = \mathcal{S}^c$
      end if
    else
      if rand[0,1] $< e^{-\Delta/T}$ **then**
        $\mathcal{S}^c = \mathcal{S}^n$
      end if
    end if
    M$'$ = M$'$ - 1
  until M$' = 0$
  T = γT; M = $\lfloor \phi$M$\rfloor$
end while
RETURN $Z(\mathcal{S}^b)$ and $\mathcal{S}^b$

As mentioned above, for a current solution, the biased branching strategy favors a direction that brings the $|\mathcal{C}|$ in a neighboring solution closer to Q. In order to implement this strategy, we define a ratio r as $|\mathcal{C}|/(|\mathcal{C}| + Q)$ which is a measure of closeness of $|\mathcal{C}|$ to Q. An r value that is greater than 0.5 implies that the value of $|\mathcal{C}|$ is more than Q, and, thus, in branching, we assign a higher probability to branching to LC-M() initiated neighborhoods. Otherwise, when r is less than 0.5, a higher probability is assigned to branching to neighborhoods employing LC-P(). In either case, we pick the better of the two solutions (in terms of objective value) on the branched side (see Figure 9). We implement this branching strategy in lines 6-11 of the SABB Algorithm given in Display 5.

III.3.3. Tabu Search with Complete Branching (TSCB)

Tabu search is another meta-heuristic framework where uphill moves are allowed during the search procedure for the purpose of escaping from local optima and, thus, exploring a larger solution space. This inevitably presents the possibility of re-visiting a solution that has been considered in previous iterations, which is called cycling. The tabu mechanism is designed to prevent cycling by storing some characteristic of the already visited solutions so that the same characteristic of a new solution can be compared against it to see if the new solution can be accepted. Details of several features of tabu search can be found in Glover and Laguna (1997) and Sait and Youssef (1999).

It is important to note that the solution representation of the problem at hand and the neighborhood function employed are intimately related to the design of the tabu mechanism. While designing the tabu mechanism, we again have to identify certain algorithmic parameters. First and foremost, we need to determine what attribute of an already visited solution will be used while forming and modifying a tabu list. We represent the tabu list as a set and call it the `TabuSet`. Given our solution representation, which uses the subsets of commodities and the compound nature of the neighborhood functions, we observe that it is difficult to determine a simple solution attribute that is effectively linked to changes in a solution in terms of the operations performed to obtain its neighboring solutions. In this case, one alternative is to store a complete solution when visited (i.e., the solution $\mathcal{S}$ with the sets $\mathcal{C}$ and $\mathcal{D}$). However, it becomes immediately clear that this approach is especially costly in terms of the number of computations required to store the data in the `TabuSet` and to verify the tabu status of a candidate solution. Incidentally, our initial numerical tests quickly illustrated the computational inefficiency of defining a tabu status based on solution representation and neighborhood function. Thus, we choose to identify the tabu status of a visited solution using a function value that can be calculated given the contents of its $\mathcal{C}$ and $\mathcal{D}$ sets. One may define several functional forms that can take a solution $\mathcal{S}$ and generate a real-valued number that represents this solution. For this purpose, instead of identifying such a new function, we simply employ the objective function of our problem. Using the objective function value as the tabu attribute has several advantages in our case. First, the `TabuSet` is simply a collection of real numbers, which can be efficiently searched to see if it contains a given value while checking the tabu status of a new solution. Secondly, since a long `TabuSet` can be kept without much computational burden, it is possible to not limit tabu tenure, i.e., the number of iterations a solution stays in `TabuSet`. This is helpful in the sense that, since we are employing a relatively large neighborhood due to compounding effects, it is possible to re-visit a solution long after its first encounter. A large `TabuSet` handles this situation efficiently by facilitating a long-term tabu memory. Furthermore, an aspiration criterion is not needed due to the nature of the `TabuSet`. Finally, we note that using a solution representation-based tabu attribute leads to prohibiting visitation of a region of the solution space that is implied by a specific tabu move (multiple solutions are usually forbidden by the same tabu attribute). On the other hand, using an objective function-based tabu criterion prohibits re-visiting individual solutions (that are likely to be represented by their unique objective function values). Thus, by using compound neighborhood functions, significant diversification characteristics are still instilled in the search.

In our tabu search algorithm, given in Display 6, we begin with an initial solution obtained from a construction heuristic. The `TabuSet` is initialized with the objective value of this solution. At each iteration of the algorithm, we generate a list of candidate solutions, `CandidateSet`, using the neighborhood function employed and the branching structure given in Figure 4. Specifically, recall that (section III.2.3) a number of neighborhood solutions are visited while using the simple neighborhoods in a compound setting, i.e., the CC component which may correspond to using CC-PN, CC-PLSN, CC-SN or CC-SLSN. Furthermore, during the local search with these compound neighborhoods (section III.2.4), CC-P() and CC-M(), several intermediate improving solutions are visited to obtain the node solutions in Figure 4. We include all of the visited solutions that provide an improvement in any iteration of these local search routines into the `CandidateSet` in a tabu search iteration. Since at each iteration of tabu search, we consider all four nodal solutions of the branching structure, we call our overall procedure a Tabu Search with Complete Branching (TSCB).

Once the `CandidateSet` is formed, the solution with the minimum objective function value becomes the current solution if its objective value is not in the `TabuSet`. Furthermore, if that solution also improves the best solution, the best solution is updated. If the best solution (with the minimum objective value) in the `CandidateSet` cannot be taken as the current solution, then it is removed from the `CandidateSet`, and the next best solution is considered to be the current solution. Each time a new solution is accepted as the current solution, we insert its objective function value into the `TabuSet`. At the end of each iteration, if the best solution available is updated, we reset the counter `nic` to zero; otherwise, it is increased by one. The purpose of `nic` is to keep track of the number of successive iterations in which the current best solution is not improved. We use `nic` along with the counter `iter` to set the stopping rule of the tabu search algorithm where the corresponding upper limits are `MAXINC` and `MAXITER`, respectively. Note that the parameter values are specified as N and 0.35 `MAXITER` for `MAXITER` and `MAXINC`, respectively, in the initialization step of Display 6. These values were obtained after some fine-tuning and used in our computational tests presented in Section III.4.

III.3.4. An Alternative Solution Representation and Search Procedure

As described in Section III.2.1, a solution may be represented as partitioning of the commodity set given by $\mathcal{S}$. One of the characteristics of such a representation is the ease by which the objective function value can be calculated. Alternatively, we could also represent a solution by direct and TL shipments installed on transfer links. Recalling the relationship between our problem and SSCFLP, this method of solution representation is directly analogous to specifying a solution in a facility location problem by fixing only the locations but not the customer assignments to these locations. Thus, the fixed TL shipments on the transfer links do not provide information about how the commodities are assigned to them. In order to evaluate the objective function value, each commodity must either be assigned to one of the TL shipments or sent by a direct shipment. This assignment problem is clearly the Generalized Assignment Problem (GAP) which is an NP-Hard problem that needs to be solved whenever the goodness of a neighborhood solution is evaluated in a heuristic solution procedure. Instead of solving GAP to optimality, one may employ some simple heuristic to find a sub-optimal solution and keep the computation time short. This approach may be inefficient because of poor solution quality. If this alternative solution representation is employed, a neighborhood function similar to the PN neighborhood function can be defined as a compounding of three simple neighborhood functions. In such an analogous framework, for the level-change stage, we can use "drop" (similar to LC-M) and "add" (similar to LC-P) neighborhoods which correspond to removing an existing TL shipment from a transfer link and adding a TL shipment on a transfer link, respectively. For the content-change stage, we can use an "exchange" (similar to CC-PN) neighborhood which adds a TL shipment to a transfer link and removes an existing TL shipment that is on some other transfer link. We implemented such an approach in our LSDB framework for small problem instances; however, the results were not encouraging when compared to the use of solution representation and a search procedure using partitioning of commodities. Thus, in the following section, we provide our extensive computational results for the latter approaches which are described in previous sections.

III.4. Computational Study

The objective of our computational study is to evaluate the performance of the combinations of our proposed compound neighborhood functions and heuristic methods on the basis of solution quality and time. We have four compound neighborhood functions differing mainly in terms of their CC components; thus, we abbreviate them using the notation PN, SN, PLSN and SLSN in general. Since we consider three heuristic approaches, namely LSDB, SABB and TSCB, using these neighborhood functions,

Display 6 TSCB Algorithm

initialize `iter`=0, `MAXITER` $=N$,
`nic`=0, `MAXNIC` $= 0.35$ `MAXITER`
$\mathcal{S}^c$ = ConstructionHeuristic(), $\mathcal{S}^b = \mathcal{S}^c$
`TabuSet` $= \{Z(\mathcal{S}^c)\}$
while `iter` $<$ `MAXITER` and `nic` $<$ `MAXNIC` **do**
  `CandidateSet` $= \emptyset$
  Find $\mathcal{S}^c_L$, $\mathcal{S}^c_{LL}$, $\mathcal{S}^c_R$, $\mathcal{S}^c_{RR}$ and form the `CandidateSet`
  `Flag` $= 0$
  repeat
    $\mathcal{S}^p = \arg\min\{Z(\mathcal{S}) : \mathcal{S} \in$ `CandidateSet`$\}$
    if $Z(\mathcal{S}^p) \notin$ `TabuSet` **then**
      $\mathcal{S}^c = \mathcal{S}^p$;
      `Flag` $= 1$
    else
      `CandidateSet` = `CandidateSet`$\setminus\{\mathcal{S}^p\}$
    end if
  until `Flag` =1 or `CandidateSet`=$\emptyset$
  if $Z(\mathcal{S}^c) < Z(\mathcal{S}^b)$ **then**
    $\mathcal{S}^b = \mathcal{S}^c$, `nic`=0
  else
    `nic` ++
  end if
  `TabuSet` = `TabuSet` $\cup\{Z(\mathcal{S}^c)\}$
  `iter` ++
end while
RETURN $Z(\mathcal{S}^b)$ and $\mathcal{S}^b$

we have a total of twelve approaches for solving our problem. For relatively small-size benchmark test instances, we compare heuristic solutions with solutions obtained using the branch-and-cut approach as implemented in CPLEX 9.0 with default settings including cut generations. On the other hand, for relatively larger test instances, we compare our approaches against each other.

III.4.1. Experimental Setup

In order to test the efficiency of the proposed solution approaches, we conduct computational experiments using randomly generated problem instances. The process of generating test instances is given in page 15. We observe that CPLEX presents limitations on the problem size that can be considered since the size of the branch-and-cut tree increases to levels that prohibit the use of memory for its storage. Thus, the size of the instances in the benchmark data set (Dataset 0) is determined accordingly in order to obtain some results for comparison purposes. Furthermore, in all of the experimental data sets we generate, we try to capture the main characteristics of the realistic problem instances as outlined below.

The rest of the input data is given in Table 1 which includes four data sets. In all of the data sets, A is set to 100, and the data sets differ from each other in terms of E, N_P, N, M and U. For each value of N, M and U, we randomly generate 10 instances. For simplicity, we choose $|\mathcal{J}| = |\mathcal{K}| = M$, which implies M^2 possible directed links for TL shipments. Dataset 0 includes 240 small instances where CPLEX provides some benchmark results in the form of either an exact solution or upper and lower bounds upon termination with a runtime limit of 2 hours. Datasets 1 through 3 have comparatively larger numbers of commodities and tighter truck capacities as given in Table 1. The instances in Dataset 1 are solvable using only the LSDB, SABB, and TSCB approaches based on all four neighborhood functions PN, PLSN, SN and SLSN.

We use this data set to compare the effectiveness of all of the 12 approaches mentioned above. Datasets 2 and 3 represent even larger problems and they are solved using LSDB, SABB and TSCB with the neighborhood function PN, only, since, based on our observations with Dataset 1, this neighborhood function appears to be the most effective in terms of runtime with only a small compromise in solution quality.

Table 1 Experimental Data Sets

Datasets	E	N_P	N	M	U	No of Instances	Remarks
Dataset 0	15	15	$25, 30, \ldots, 60$	4, 6, 8	8	240	CPLEX LSDB, SABB, TSCB PN, PLSN, SN, SLSN
Dataset 1	15	15	$100, 110, \ldots, 150$	4	4, 6, 8	180	LSDB, SABB, TSCB PN, PLSN, SN, SLSN
Dataset 2	25	25	$300, 330, \ldots, 450$	4	4, 6, 8	180	LSDB, SABB, TSCB PN
Dataset 3	35	30	$500, 550, \ldots, 750$	4	4, 6, 8	180	LSDB, SABB, TSCB PN

III.4.2. Computational Results

As described in Section III.2.2, we have two methods, C-RC() and C-RL(), for constructing initial feasible solutions. We performed tests to compare the relative performance of these methods and determined that, on average, C-RC() provides initial solutions with better quality. Furthermore, we tested these construction methods with each of the three heuristic approaches (LSDB, SABB and TSCB), and we obtained better results with the LSDB when the initial solutions were obtained by using C-RC() and with SABB and TSCB when initialized with C-RL(). Therefore, when generating initial solutions for all of the numerical tests described below, we employ C-RC() with LSDB and C-RL() with SABB and TSCB.

For exact solutions of the benchmark instances (Dataset 0), summarized in Table 2, we employ a stopping criteria of 2 hours time limit for CPLEX. Upon termination, we record the runtime, the best lower bound (Z_{LB}) and the best upper bound (Z_{UB}), and calculate the percentage optimality gap as $100 \times (Z_{UB} - Z_{LB})/Z_{LB}$. In Table 2, for each value of N and M, we present the average and maximum optimality gaps as well as the solution times over 10 instances. These results clearly illustrate the computational difficulties even with small sized problems.

Table 2 Summary of Exact Solution Results (Dataset 0 solved with CPLEX)

N	$M=4$				$M=6$				$M=8$			
	% Gap		Time (Sec.)		% Gap		Time (Sec.)		% Gap		Time (Sec.)	
	Ave	Max	Ave	Max	Ave	Max	Ave	Max	Ave	Max	Ave	Max
25	0.01	0.01	30.1	251.0	0.54	5.31	732.1	7201.0	1.29	12.82	778.0	7201.0
30	0.01	0.01	4.5	15.0	0.01	0.01	73.9	221.0	0.01	0.01	185.2	589.0
35	2.80	6.13	5837.9	7204.0	3.40	7.48	6024.1	7201.0	3.13	9.90	5791.3	7200.0
40	5.12	11.50	5550.6	7201.0	5.51	11.74	6136.9	7201.0	8.40	13.06	7069.2	7200.0
45	1.79	9.60	3696.3	7204.0	2.41	7.45	5882.1	7200.0	2.07	8.64	5901.8	7200.0
50	2.43	6.99	6607.6	7205.0	2.97	8.65	6907.7	7200.0	3.23	10.46	6579.3	7200.0
55	5.32	7.84	6608.5	7201.0	6.28	12.32	7200.0	7200.0	7.66	11.74	7200.3	7203.0
60	4.51	9.08	7200.2	7201.0	5.39	11.25	7200.0	7200.0	6.61	11.39	7200.3	7202.0
Ave	2.75	6.39	4442.0		3.31	8.03	5019.6		4.05	9.75	5088.2	

Tables 3 and 4 gives percentage gaps between the best lower bounds obtained with CPLEX and the heuristic approaches with varying neighborhood functions. Table 3 and 4 have total four subtables embedded, one for each compound neighborhood function, PN, PLSN, SN and SLSN. We use average and maximum percentage gap and average runtime for comparison. A percentage gap is calculated as $100 \times (Z_{heur} - Z_{LB})/Z_{LB}$ where Z_{heur} is the objective function value of the appropriate heuristic solution. For example, considering the use of the PN neighborhood with $M = 4$, LSDB, SABB and TSCB provide solutions with overall average gaps of 3.53%, 3.66% and 3.86%, respectively, whereas the corresponding average gap for CPLEX is 2.75%. The maximum gaps for these metaheuristics are 7.35%, 7.50% and 7.93%, respectively, while the corresponding maximum gap for CPLEX is 6.39%. A similar trend is observed in the case of $M = 6$ and $M = 8$; thus, the quality of the solutions obtained by our algorithms is comparable to to those obtained using CPLEX. The solution quality obtained by our algorithms utilizing the PLSN, SN and SLSN neighborhood functions also confirms that LSDB, SABB and TSCB provide good quality solutions. Given the uncompromising solution quality of our approaches, it is clear that the main advantage of the new algorithms is their effectiveness in terms of solution time. The runtime for all of the heuristic methods is significantly less than the runtime with CPLEX. For example, the overall average runtime with $M = 4$ is 4442.0 seconds for CPLEX and 1.61, 5.20, and 0.86 seconds when the neighborhood function PN is employed with LSDB, SABB, and TSCB, respectively. The solution times with the neighborhood functions PLSN, SN, and SLSN are also significantly lower than the CPLEX solution times; however, they are longer than those obtained when PN is employed. We note that this is an expected outcome since these latter neighborhood functions spend more time with local search routines while generating neighboring solutions.

Since CPLEX upper bounds and our heuristic approaches both provide feasible solutions, we also examine the percentage gaps between these solutions. These results are reported in Table 5. In this case, a percentage gap is calculated as $100 \times (Z_{heur} - Z_{UB})/Z_{UB}$. It is easily observed that, in the majority of cases spanning all of the heuristic approaches and neighborhood functions, the average gaps are less than 1.0%, including several instances in which the heuristic approaches find better feasible solutions as indicated by the negative percentage gap values.

Table 3 Gaps from CPLEX LB and Runtime for PLSN and SLSN (Dataset 0)

N	Ave % Gap			Max % Gap			Ave Time (Sec)		
	LSDB	SABB	TSCB	LSDB	SABB	TSCB	LSDB	SABB	TSCB
	PLSN								
	$M = 4$								
25	0.28	0.80	0.75	0.90	1.59	1.30	1.20	4.00	0.00
30	0.20	0.46	0.56	0.75	1.17	0.85	2.30	2.70	0.00
35	3.25	3.96	3.63	6.50	6.65	6.93	6.60	7.70	1.50
40	5.62	5.93	5.66	11.41	11.39	11.22	11.70	14.20	2.40
45	2.18	2.46	2.56	9.85	11.01	11.01	13.30	24.80	3.50
50	2.87	3.43	3.10	6.69	6.89	7.05	20.50	25.70	4.40
55	5.75	6.07	5.89	8.14	8.06	8.53	42.10	63.70	11.70
60	4.97	5.26	5.17	9.55	9.38	9.31	40.00	55.80	12.50
Ave.	3.14	3.55	3.42	6.72	7.02	7.02	17.21	24.83	4.50
	$M = 6$								
25	0.91	1.38	0.93	6.44	6.72	5.53	1.80	4.90	0.00
30	0.12	0.43	0.44	0.45	0.82	0.82	4.10	3.50	0.40
35	3.69	4.56	4.06	8.18	7.91	8.44	8.90	14.70	2.20
40	6.06	6.49	6.41	12.45	12.09	13.09	17.70	26.40	4.50
45	2.89	3.25	3.31	8.22	8.19	8.92	20.20	34.30	3.30
50	3.36	3.93	3.86	8.51	8.65	9.05	23.80	33.30	5.90
55	6.49	6.89	6.69	11.01	10.95	11.48	46.40	59.00	22.40
60	5.84	5.97	6.03	10.93	11.78	11.84	58.80	73.80	16.00
Ave.	3.67	4.11	3.96	8.27	8.39	8.65	22.71	31.24	6.84
	$M = 8$								
25	1.63	2.01	1.56	13.02	12.98	13.34	3.70	7.50	0.20
30	0.24	0.48	0.48	0.64	0.86	0.80	5.50	13.90	0.70
35	3.35	4.40	3.74	9.97	10.12	9.92	14.00	20.90	3.60
40	8.50	8.92	8.63	13.01	12.96	12.95	22.80	23.80	5.30
45	2.41	2.79	2.89	9.08	9.27	9.52	24.40	46.60	4.40
50	3.16	3.77	3.47	8.66	8.89	9.12	46.60	72.10	10.30
55	6.82	7.12	7.06	10.54	10.42	10.64	69.80	96.00	31.30
60	6.16	6.36	6.30	9.69	9.35	10.35	89.90	152.10	35.50
Ave.	4.03	4.48	4.26	9.33	9.36	9.58	34.59	54.11	11.41
	SLSN								
	$M = 4$								
25	0.52	0.69	0.64	1.08	1.59	1.30	1.70	8.20	0.00
30	0.28	0.43	0.40	0.75	1.17	0.85	2.50	13.10	0.00
35	3.47	3.47	3.82	6.56	6.65	6.93	10.00	22.10	1.10
40	5.74	5.37	6.03	11.48	11.39	11.22	18.80	56.30	1.60
45	2.39	2.37	2.50	10.74	11.01	11.01	16.50	47.60	1.70
50	2.88	2.90	3.33	6.74	6.89	7.05	42.30	75.70	4.20
55	5.86	5.55	6.09	8.18	8.06	8.53	64.20	123.50	9.40
60	5.10	4.90	5.06	9.68	9.38	9.31	64.40	106.40	7.40
Ave.	3.28	3.21	3.48	6.90	7.02	7.02	27.55	56.61	3.18
	$M = 6$								
25	1.11	1.13	1.03	5.53	6.72	5.53	4.20	10.00	0.10
30	0.13	0.28	0.45	0.33	0.82	0.82	4.40	17.60	0.00
35	3.99	3.75	4.22	8.48	7.91	8.44	15.60	36.30	1.70
40	6.25	5.98	6.28	12.67	12.09	13.09	26.10	70.70	3.10
45	3.09	2.82	3.22	8.34	8.19	8.92	22.60	67.60	1.30
50	3.47	3.24	3.51	8.69	8.65	9.05	48.80	88.10	5.70
55	6.53	6.36	6.73	10.80	10.95	11.48	70.60	117.30	13.10
60	5.88	5.74	6.17	10.87	11.78	11.84	79.80	138.60	9.30
Ave	3.81	3.67	3.95	8.21	8.39	8.65	34.01	68.28	4.29
	$M = 8$								
25	1.70	1.79	1.68	13.15	12.98	13.34	4.70	14.10	0.10
30	0.15	0.28	0.30	0.55	0.86	0.80	4.70	17.60	0.10
35	3.49	3.43	3.58	10.07	10.12	9.92	32.30	75.30	2.80
40	8.85	8.45	8.61	13.03	12.96	12.95	33.80	84.80	3.50
45	2.81	2.43	2.78	9.50	9.27	9.52	40.70	113.00	2.70
50	3.33	3.19	3.61	8.91	8.89	9.12	114.30	196.80	8.90
55	6.93	6.65	7.03	10.43	10.42	10.64	118.00	192.90	20.50
60	6.34	5.90	6.36	10.53	9.35	10.35	123.60	200.00	12.70
Ave	4.20	4.01	4.25	9.52	9.36	9.58	59.01	111.81	6.41

Table 4 Gaps from CPLEX LB and Runtime for PN and SN (Dataset 0)

N	Ave % Gap			Max % Gap			Ave Time (Sec)		
	LSDB	SABB	TSCB	LSDB	SABB	TSCB	LSDB	SABB	TSCB
	PN								
	$M = 4$								
25	0.54	0.91	1.10	0.99	1.91	1.80	0.00	1.20	0.00
30	0.27	0.56	0.55	0.75	1.08	1.17	0.10	1.80	0.00
35	3.63	4.06	4.31	7.02	7.94	9.07	2.20	1.80	0.30
40	6.01	6.03	6.18	12.06	11.52	11.49	0.90	2.60	0.60
45	2.88	2.62	2.71	10.81	11.01	11.01	0.50	7.30	0.40
50	3.31	3.52	3.99	7.41	7.77	8.50	2.80	5.10	0.60
55	6.13	6.29	6.58	9.05	8.63	9.91	3.90	9.80	2.90
60	5.46	5.30	5.46	10.74	10.11	10.53	2.44	12.00	2.10
Ave	3.53	3.66	3.86	7.35	7.50	7.93	1.61	5.20	0.86
	$M = 6$								
25	1.14	3.99	1.69	5.53	14.47	6.98	0.00	1.80	0.00
30	0.12	0.50	0.34	0.23	0.96	0.77	0.10	1.50	0.00
35	4.10	4.62	4.77	8.34	8.66	9.01	2.60	2.40	0.40
40	6.75	6.34	7.17	13.01	13.32	13.04	1.10	6.90	1.30
45	3.57	3.17	3.38	8.60	8.30	8.34	0.70	5.90	0.40
50	3.98	3.98	4.38	8.68	8.64	9.16	3.10	5.90	1.50
55	6.88	6.80	7.25	11.40	11.19	12.22	4.50	9.50	4.30
60	6.44	6.28	6.40	11.61	11.57	11.79	3.00	13.70	2.60
Ave	4.12	4.46	4.42	8.42	9.64	8.91	1.89	5.95	1.31
	$M = 8$								
25	1.73	1.96	2.04	13.15	13.34	13.26	0.20	2.50	0.00
30	0.24	0.60	0.60	0.85	1.38	1.16	0.60	4.50	0.10
35	3.79	4.36	4.45	10.64	10.73	10.18	4.50	3.20	0.90
40	9.04	9.24	9.37	13.24	13.30	14.65	1.60	5.70	1.60
45	3.25	3.00	2.99	9.58	9.78	9.61	1.00	10.10	1.00
50	4.07	3.81	4.44	9.46	9.19	9.84	3.80	9.40	3.30
55	7.40	7.33	8.15	11.33	12.13	13.52	5.80	13.30	5.30
60	6.89	6.39	7.06	10.65	10.14	15.45	3.60	19.90	4.40
Ave	4.55	4.59	4.89	9.86	10.00	10.96	2.64	8.58	2.08
	SN								
	$M = 4$								
25	0.36	0.73	0.40	1.13	1.38	1.62	1.00	2.40	0.00
30	0.26	0.81	0.48	0.75	2.33	0.99	0.70	1.40	0.00
35	3.44	3.54	3.56	6.74	6.62	6.80	4.50	18.70	1.20
40	5.57	5.50	5.74	11.48	11.45	11.38	6.10	17.50	1.50
45	2.32	2.57	2.74	10.19	11.01	11.01	4.00	13.00	0.60
50	2.94	2.97	3.18	6.87	7.28	7.21	13.20	32.10	4.70
55	5.80	5.72	5.91	8.09	8.49	8.70	16.40	47.70	8.50
60	4.94	5.13	5.30	9.43	9.27	10.40	12.30	29.00	4.80
Ave	3.20	3.37	3.41	6.84	7.23	7.26	7.28	20.23	2.66
	$M = 6$								
25	0.94	1.34	0.62	6.14	6.72	6.52	1.70	8.20	0.00
30	0.19	0.38	0.47	0.59	0.98	1.21	0.90	3.10	0.00
35	3.86	3.85	4.06	8.62	8.22	8.33	5.80	19.70	1.90
40	6.06	6.04	6.42	12.37	12.50	12.42	8.10	33.30	2.80
45	2.95	3.07	3.42	8.56	8.19	8.93	5.10	13.80	1.20
50	3.41	3.53	3.52	8.63	8.60	8.25	12.80	29.50	5.90
55	6.50	6.43	6.58	11.06	10.66	11.04	17.00	42.40	6.60
60	5.77	5.90	6.15	10.99	11.48	11.45	15.10	22.70	4.20
Ave	3.71	3.82	3.91	8.37	8.42	8.52	8.31	21.59	2.83
	$M = 8$								
25	1.57	1.93	1.04	12.98	13.70	13.70	2.80	7.70	0.10
30	0.21	0.47	0.70	0.59	0.82	1.45	1.00	5.80	0.00
35	3.45	3.56	3.67	10.01	9.91	10.35	9.90	44.70	2.70
40	8.61	8.63	8.70	13.01	12.84	13.22	10.10	26.70	2.70
45	2.48	2.76	2.91	9.02	9.82	9.01	6.60	13.90	2.10
50	3.13	3.48	3.42	9.01	8.98	8.63	26.10	150.90	14.60
55	6.88	6.82	7.07	10.74	10.60	10.70	27.50	70.90	12.40
60	6.15	6.25	6.09	9.96	9.69	9.57	22.20	53.10	13.70
Ave	4.06	4.24	4.20	9.41	9.54	9.58	13.28	46.71	6.04

Table 5 Percentage Gaps from CPLEX Upper Bound (Dataset 0)

N	Ave. % Gap			Max % Gap			Ave. % Gap			Max % Gap		
	LSDB	SABB	TSCB	LSDB	SABB	TSCB	LSDB	SABB	TSCB	LSDB	SABB	TSCB
	PN						PLSN					
	$M = 4$											
25	0.53	0.90	1.10	0.99	1.90	1.80	0.28	0.79	0.74	0.90	1.35	1.61
30	0.27	0.56	0.54	0.74	1.08	1.16	0.20	0.45	0.56	0.74	1.16	1.16
35	0.81	1.23	1.47	1.81	2.98	3.77	0.44	1.13	0.82	0.72	2.59	1.60
40	0.86	0.87	1.03	1.69	1.83	1.87	0.48	0.79	0.52	0.92	1.47	1.49
45	1.07	0.81	0.91	1.67	1.39	1.28	0.38	0.66	0.76	0.80	1.37	1.48
50	0.87	1.08	1.53	1.69	2.81	3.03	0.43	0.98	0.66	1.00	1.73	1.77
55	0.78	0.93	1.20	1.57	1.96	2.30	0.41	0.71	0.55	0.98	1.82	1.14
60	0.91	0.76	0.92	1.77	1.53	1.90	0.45	0.72	0.64	1.19	1.57	1.59
Ave	0.76	0.89	1.09	1.49	1.94	2.14	0.38	0.78	0.65	0.91	1.63	1.48
	$M = 6$											
25	0.60	3.45	1.15	1.41	14.46	3.20	0.36	0.84	0.39	1.07	2.21	1.21
30	0.12	0.49	0.33	0.22	0.95	0.76	0.11	0.43	0.44	0.44	1.48	0.82
35	0.68	1.19	1.34	1.73	2.37	2.23	0.28	1.13	0.65	0.66	2.36	2.59
40	1.19	0.80	1.61	1.94	2.22	3.23	0.54	0.94	0.86	1.24	1.96	1.45
45	1.13	0.74	0.95	1.50	1.61	1.77	0.47	0.82	0.88	0.76	2.07	1.53
50	0.99	0.98	1.38	2.37	1.79	3.40	0.38	0.93	0.87	0.72	2.08	2.24
55	0.59	0.50	0.92	1.44	1.33	1.80	0.21	0.59	0.40	1.00	1.40	1.49
60	0.99	0.86	0.96	1.36	1.53	2.03	0.43	0.54	0.61	0.88	1.44	1.67
Ave	0.79	1.13	1.08	1.49	3.28	2.30	0.35	0.78	0.64	0.85	1.87	1.62
	$M = 8$											
25	0.44	0.66	0.74	0.92	1.38	1.45	0.34	0.72	0.27	1.00	1.24	0.71
30	0.24	0.60	0.60	0.84	1.37	1.15	0.23	0.47	0.48	0.64	0.90	1.44
35	0.65	1.21	1.30	1.55	3.05	2.48	0.22	1.25	0.61	0.74	2.52	2.23
40	0.61	0.80	0.92	1.14	1.83	2.84	0.10	0.50	0.22	0.61	1.39	0.89
45	1.16	0.91	0.91	1.90	2.10	2.02	0.33	0.70	0.80	0.63	1.66	1.82
50	0.84	0.59	1.21	2.03	1.97	2.83	-0.04	0.56	0.26	0.81	2.74	1.23
55	-0.22	-0.29	0.48	1.38	1.06	4.50	-0.76	-0.49	-0.54	0.24	0.40	0.46
60	0.27	-0.20	0.40	1.24	1.26	4.36	-0.41	-0.24	-0.29	0.96	0.38	1.20
Ave	0.50	0.54	0.82	1.38	1.75	2.70	0.00	0.43	0.23	0.70	1.41	1.25
	SN						SLSN					
	$M = 4$											
25	0.35	0.72	0.39	1.13	1.37	1.61	0.52	0.69	0.63	1.08	1.59	1.30
30	0.25	0.81	0.47	0.74	2.33	0.98	0.28	0.42	0.39	0.74	1.16	0.84
35	0.62	0.72	0.74	1.55	1.28	1.16	0.66	0.65	0.99	1.19	1.22	1.82
40	0.44	0.37	0.60	0.87	0.87	1.66	0.61	0.24	0.88	1.23	0.75	1.95
45	0.53	0.77	0.94	0.95	1.28	1.49	0.59	0.57	0.70	1.06	1.38	1.28
50	0.51	0.53	0.74	1.13	0.93	1.77	0.44	0.47	0.89	0.99	1.78	1.85
55	0.46	0.38	0.56	1.25	1.27	1.46	0.51	0.22	0.73	1.00	0.75	1.31
60	0.42	0.60	0.76	0.82	1.53	1.32	0.56	0.37	0.53	1.81	1.10	1.19
Ave	0.45	0.61	0.65	1.06	1.36	1.43	0.52	0.45	0.72	1.14	1.22	1.44
	$M = 6$											
25	0.40	0.80	0.35	0.78	1.50	1.29	0.57	0.59	0.49	1.26	1.65	1.31
30	0.18	0.37	0.46	0.58	0.97	1.20	0.12	0.28	0.45	0.32	0.82	0.82
35	0.45	0.44	0.64	1.07	1.10	1.52	0.58	0.35	0.80	1.41	0.97	1.44
40	0.53	0.52	0.89	1.00	1.18	2.34	0.72	0.47	0.74	1.96	0.98	2.23
45	0.52	0.64	0.98	1.06	1.74	1.70	0.67	0.40	0.79	1.66	1.00	1.50
50	0.43	0.55	0.54	0.76	1.46	1.46	0.49	0.27	0.53	1.20	0.59	1.19
55	0.23	0.16	0.31	0.75	1.19	1.05	0.26	0.09	0.44	0.87	0.88	1.08
60	0.36	0.49	0.73	0.85	1.26	1.57	0.47	0.34	0.74	1.52	0.68	1.96
Ave	0.39	0.50	0.61	0.86	1.30	1.52	0.48	0.35	0.62	1.28	0.95	1.44
	$M = 8$											
25	0.28	0.63	0.34	0.61	2.28	1.61	0.40	0.49	0.39	1.04	1.11	0.61
30	0.20	0.47	0.70	0.58	0.81	1.44	0.15	0.27	0.29	0.54	0.85	0.79
35	0.32	0.43	0.53	1.11	1.11	1.16	0.37	0.30	0.45	1.45	1.08	1.44
40	0.21	0.23	0.30	0.99	0.84	1.37	0.44	0.06	0.21	1.12	0.59	1.00
45	0.40	0.68	0.83	0.97	1.54	2.38	0.73	0.35	0.70	2.05	0.71	1.73
50	-0.07	0.27	0.21	0.75	1.67	1.17	0.13	-0.01	0.40	1.39	1.17	1.39
55	-0.71	-0.76	-0.54	0.74	0.86	0.34	-0.66	-0.92	-0.57	1.44	0.13	0.29
60	-0.43	-0.33	-0.48	0.60	0.68	1.07	-0.25	-0.66	-0.23	1.01	0.85	1.16
Ave	0.03	0.20	0.24	0.79	1.22	1.32	0.16	-0.01	0.21	1.26	0.81	1.05

Based on the results in Tables 3, 4 and 5, we conclude that the performance in solution quality with all four neighborhood functions PN, SN, SLSN and PLSN are comparable in terms of average and maximum percentage gaps for all three heuristics. The solution quality with the latter three neighborhood functions, in general, appears to be on the better side as indicated by the lower percentage gaps in Tables 3, 4 and 5. On the other hand, in terms of solution runtime, regardless of the heuristic approach, use of the neighborhood function PN causes algorithms to perform much faster than the use of SN, SLSN, and PLSN.

Next, we solve the problem instances in Dataset 1 using the LSDB, SABB and TSCB algorithms, each with PN, PLSN, SN and SLSN neighborhood functions, i.e., each problem instance is solved by 12 different combinations of four compound neighborhoods (PN, PLSN, SN and SLSN) and three heuristic algorithms (LSDB, SABB and TSCB). In Table 6, in the columns under the heading No of Times Best, we report the number of instances a specific algorithm-neighborhood function combination finds the best solution.

For example, out of 10 instances for problem class with $N = 110$ and $U = 4$, LSDB implemented with the PN neighborhood function finds the best solution in one instance. In the second group of three columns for the methods, we provide the average of the percentage gaps from the best obtained solution (out of twelve solutions) over all 10 instances of a problem class. Obviously, lower percentages indicate better performance of a method-neighborhood function combination in providing a good quality solution.

Table 6 Comparison of Algorithms and Neighborhood Functions (Dataset 1)

		N	No. of Times Best			Ave % Gap			Ave Time (Sec.)		
			LSDB	SABB	TSCB	LSDB	SABB	TSCB	LSDB	SABB	TSCB
$U = 4$	PN	100	0	0	0	0.81	0.63	0.62	32.40	40.30	19.20
		110	1	0	0	1.33	0.52	0.62	51.60	48.30	20.60
		120	0	1	0	1.15	0.46	0.66	49.40	59.50	22.10
		130	1	0	0	1.63	0.62	0.70	65.80	73.60	48.50
		140	0	0	0	1.44	0.48	0.57	62.20	84.40	54.80
		150	0	0	0	1.08	0.45	0.46	96.50	96.90	62.30
	PLSN	100	0	1	1	0.43	0.40	0.26	264.40	249.10	167.40
		110	0	0	0	0.45	0.47	0.32	392.80	309.90	377.20
		120	1	1	0	0.36	0.48	0.38	480.90	448.40	364.60
		130	0	1	2	0.57	0.61	0.33	601.70	555.40	660.90
		140	0	0	1	0.62	0.49	0.21	738.30	622.50	635.40
		150	0	0	2	0.47	0.45	0.16	976.80	881.00	1245.70
	SN	100	1	1	1	0.46	0.37	0.41	79.50	95.30	53.90
		110	0	3	1	0.32	0.13	0.26	124.70	182.80	109.60
		120	1	1	0	0.31	0.28	0.35	133.50	168.50	96.50
		130	0	1	0	0.52	0.29	0.37	197.50	237.70	204.10
		140	0	0	1	0.48	0.29	0.36	204.90	273.20	180.50
		150	0	2	2	0.42	0.18	0.21	261.70	334.70	369.80
	SLSN	100	2	3	0	0.38	0.14	0.43	126.30	191.00	81.20
		110	0	4	1	0.43	0.10	0.33	195.50	421.60	238.80
		120	1	3	1	0.30	0.17	0.34	214.30	348.50	126.80
		130	1	2	2	0.50	0.28	0.27	302.80	499.10	284.10
		140	2	4	2	0.46	0.09	0.42	359.50	493.70	389.40
		150	0	4	0	0.43	0.13	0.26	459.90	816.40	513.30
$U = 6$	PN	100	7	0	0	0.32	1.33	1.48	32.40	38.80	12.30
		110	1	0	0	1.37	0.79	1.13	51.60	52.00	21.90
		120	2	1	0	1.09	0.55	0.78	49.40	46.00	32.30
		130	3	0	0	1.50	0.88	1.12	65.80	65.10	32.90
		140	2	0	0	0.79	0.57	0.65	62.20	74.70	48.30
		150	2	0	0	0.91	0.68	0.81	96.50	106.30	67.50
	PLSN	100	2	0	0	0.92	1.12	1.03	525.30	509.00	219.90
		110	0	0	1	0.49	0.62	0.28	654.40	402.40	377.90
		120	1	1	0	0.29	0.57	0.61	817.30	705.70	273.60
		130	2	1	0	0.61	0.93	0.84	1180.80	900.90	515.00
		140	2	0	1	0.41	0.56	0.22	1364.20	829.90	821.30
		150	0	0	0	0.58	0.71	0.47	1860.00	1517.00	961.90
	SN	100	0	0	1	1.03	0.80	0.99	69.30	135.80	73.40
		110	0	8	0	0.40	0.12	0.36	136.40	307.10	157.20
		120	0	1	0	0.28	0.34	0.50	128.40	219.30	127.90
		130	2	1	0	0.59	0.69	0.78	169.10	238.10	233.10
		140	0	2	0	0.35	0.30	0.40	184.60	334.20	175.50
		150	1	3	2	0.48	0.40	0.41	264.40	520.40	347.30
	SLSN	100	0	0	0	1.00	0.86	1.02	119.60	240.20	95.40
		110	0	0	0	0.50	0.25	0.51	215.30	556.00	201.50
		120	1	3	0	0.26	0.29	0.34	207.30	445.80	173.30
		130	1	1	0	0.61	0.50	0.75	293.80	561.40	253.80
		140	2	1	0	0.34	0.27	0.33	331.90	707.30	505.20
		150	0	1	1	0.53	0.47	0.47	478.60	1147.00	524.10
$U = 8$	PN	100	0	0	0	0.47	0.64	0.96	40.60	37.10	22.30
		110	0	0	0	0.35	0.51	0.65	41.80	53.50	17.10
		120	1	0	0	0.41	0.79	0.73	55.00	53.70	45.10
		130	0	0	0	0.38	0.77	0.91	111.30	67.80	51.70
		140	1	0	0	0.57	0.58	0.64	73.80	83.80	50.50
		150	0	0	0	0.49	0.73	0.81	115.60	95.50	86.50
	PLSN	100	1	0	0	0.36	0.69	0.34	299.20	200.50	225.20
		110	0	0	1	0.32	0.52	0.26	419.00	739.00	390.20
		120	0	1	0	0.38	0.51	0.53	571.50	871.10	663.80
		130	1	0	3	0.25	0.78	0.42	703.20	671.10	519.60
		140	1	1	1	0.33	0.50	0.30	912.40	996.00	1119.50
		150	1	0	0	0.31	0.58	0.35	1080.10	1002.00	1111.20
	SN	100	1	2	2	0.29	0.22	0.28	88.00	197.50	67.90
		110	5	3	0	0.21	0.19	0.48	82.20	206.40	142.50
		120	1	1	2	0.37	0.30	0.27	119.20	250.50	160.70
		130	1	3	2	0.23	0.26	0.47	207.90	513.20	204.10
		140	0	1	1	0.33	0.37	0.29	178.60	310.20	287.60
		150	0	3	4	0.27	0.12	0.14	278.40	558.10	507.50
	SLSN	100	0	4	0	0.36	0.14	0.43	156.60	307.30	114.00
		110	0	0	1	0.29	0.22	0.24	142.70	502.50	158.50
		120	1	2	1	0.38	0.37	0.40	198.30	463.10	189.20
		130	0	0	0	0.40	0.34	0.49	340.30	805.50	395.30
		140	2	2	0	0.28	0.26	0.35	358.90	782.80	459.60
		150	0	2	0	0.24	0.14	0.27	505.40	1158.50	648.40

Finally, in the last portion of Table 6, we report the average solution times over 10 instances of each problem class. A compact summary of the results in Table 6 is presented in Table 7 in which we report the percentages of the number of times an algorithm-neighborhood function combination provides the best solution. For example, for the case where $U = 4$, the LSDB with PN neighborhood gives the best solution in 2 instances (1 instance for $N = 110$ and 1 instance for $N = 130$) out of 60 instances (10 instances for each of six problem classes). Thus, the associated percentage in the third column of Table 7 is 3.33%. In the last six columns of Table 7, we present the averages (over six problem classes where the N values vary) of corresponding entries in Table 6. The results in these tables show that the neighborhood function PLSN, regardless of the heuristic algorithm it is used with, generates the least desirable solutions since the solution times are always the highest and the solution quality does not appear significantly better compared to other neighborhood functions. In most cases, the solution quality is slightly worse than the solution qualities obtained via the use of the SN and SLSN neighborhoods, and we obtain better solution time performance with these latter neighborhoods. Furthermore, we observe that, when the PN neighborhood function is employed, the solution times are significantly better with each of the three heuristic algorithms and, at the same time, the solution qualities are at easily acceptable levels as illustrated by the majority of the results where the average percentage gaps are below 1.0%. On the other hand, from the heuristic algorithms perspective, SABB appears to perform the best most of the time as it finds the best solutions most of the time, and, thus, provides lower average percentage gaps. Especially, the combination of SABB and the neighborhood functions SN or SLSN perform exceptionally well in terms of solution quality as measured by the number of times they provide the best solution, but not in terms of solution times as noted above.

Table 7 Summary of Comparisons (Dataset 1)

		% No. of Times Best			Ave % Gap			Ave Time (Sec.)		
		LSDB	SABB	TSCB	LSDB	SABB	TSCB	LSDB	SABB	TSCB
$U = 4$	PN	3.33	1.67	0.00	1.24	0.53	0.61	59.65	67.17	37.92
	PLSN	1.67	5.00	10.00	0.48	0.48	0.28	575.82	511.05	575.20
	SN	3.33	13.33	8.33	0.42	0.26	0.33	166.97	215.37	169.07
	SLSN	10.00	33.33	10.00	0.42	0.15	0.34	276.38	461.72	272.27
$U = 6$	PN	28.33	1.67	0.00	1.00	0.80	0.99	59.65	63.82	35.87
	PLSN	11.67	3.33	3.33	0.55	0.75	0.57	1067.00	810.82	528.27
	SN	5.00	25.00	5.00	0.52	0.44	0.57	158.70	292.48	185.73
	SLSN	6.67	10.00	1.67	0.54	0.44	0.57	274.42	609.62	292.22
$U = 8$	PN	3.33	0.00	0.00	0.45	0.67	0.79	73.02	65.23	45.53
	PLSN	6.67	3.33	8.33	0.33	0.60	0.36	664.23	746.62	671.58
	SN	13.33	21.67	18.33	0.28	0.24	0.32	159.05	339.32	228.38
	SLSN	5.00	16.67	3.33	0.33	0.25	0.36	283.70	669.95	327.50

Since, based on the Dataset 1 results, we observe that the adoption of our proposed compound neighborhood functions lead to faster algorithms without serious compromise in solution quality, in the last stage of our numerical study, we concentrate on comparing only the heuristic algorithms while employing the neighborhood function PN in each one. For this purpose of comparing LSDB, SABB, and TSCB, we consider the even larger problem instances provided by Datasets 2 and 3. The results are summarized in Table 8. In this table, we observe that the TSCB performs best in terms of solution quality (average percentage gaps) with the largest instances given by Dataset 3. We also observe that the TSCB and SABB algorithms present very similar performance when Dataset 2 is used.

In Dataset 3, SABB does not perform as well as TSCB; however, its performance is also very satisfactory as it provides less than a 0.5% gap on average from the best available solution. Furthermore, solution times with SABB are highly favorable for large instances in Dataset 3 when compared to the solution times with LSDB and TSCB. Thus, for practical problems of large size, it appears that the SABB algorithm, implemented by employing the neighborhood function PN, is a promising approach both in terms of solution quality and solution time.

Table 8 Comparing LSDB, SABB and TSCB using the Neighborhood Function PN (Datasets 2 and 3)

		No. of Times Best			Ave % Gap			Ave Time (Sec.)		
	N	LSDB	SABB	TSCB	LSDB	SABB	TSCB	LSDB	SABB	TSCB
	Dataset 2									
$U=4$	300	0	9	1	0.96	0.02	0.23	538.00	772.50	497.70
	330	0	6	4	1.03	0.10	0.09	664.27	818.91	593.18
	360	0	6	4	1.14	0.12	0.15	875.20	1025.20	829.20
	390	0	5	5	0.90	0.10	0.09	867.70	1110.70	1285.70
	420	0	6	4	0.87	0.10	0.08	1233.60	1287.10	1494.50
	450	0	5	5	1.09	0.14	0.09	1668.70	2236.00	2079.20
	Ave.	0.00	6.17	3.83	1.00	0.10	0.12	974.58	1208.40	1129.91
$U=6$	300	2	6	2	0.51	0.16	0.39	643.80	693.30	444.70
	330	3	3	4	0.49	0.18	0.10	640.90	730.90	640.00
	360	2	4	4	0.56	0.16	0.15	917.10	985.00	745.20
	390	1	5	4	0.56	0.22	0.33	808.90	1075.50	1141.00
	420	3	1	6	0.54	0.24	0.06	1183.80	1054.30	1141.60
	450	0	6	4	0.74	0.15	0.15	1796.60	1895.20	2084.80
	Ave.	1.83	4.17	4.00	0.57	0.18	0.19	998.52	1072.37	1032.88
$U=8$	300	3	3	4	0.41	0.21	0.34	651.90	764.00	568.70
	330	8	1	1	0.16	0.41	0.49	852.90	652.70	889.50
	360	4	4	2	0.39	0.23	0.33	1148.60	867.20	1274.20
	390	3	3	4	0.14	0.25	0.19	1038.20	1045.40	1031.30
	420	3	1	6	0.26	0.32	0.11	1714.30	1011.70	1309.20
	450	3	5	2	0.34	0.26	0.26	2138.20	1537.50	1508.40
	Ave.	4.00	2.83	3.17	0.28	0.28	0.29	1257.35	979.75	1096.88
	Dataset 3									
$U=4$	500	0	1	9	0.53	0.46	0.01	2296.50	956.10	3663.20
	550	0	3	7	0.59	0.28	0.05	2610.30	1604.50	2668.80
	600	0	4	6	0.71	0.17	0.09	3742.80	2775.40	5226.00
	650	1	3	6	0.67	0.21	0.12	4713.50	3538.60	10265.20
	700	0	1	9	0.84	0.23	0.01	7560.70	3281.60	11152.40
	750	0	4	6	0.57	0.21	0.07	7694.90	5718.60	8912.00
	Ave.	0.17	2.67	7.17	0.65	0.26	0.06	4769.78	2979.13	6981.27
$U=6$	500	2	4	4	0.29	0.38	0.09	1963.40	919.30	2876.50
	550	2	2	6	0.52	0.22	0.15	2398.60	2487.70	3068.30
	600	0	3	7	0.57	0.24	0.03	3701.30	2394.60	5182.10
	650	1	3	6	0.43	0.20	0.05	4626.60	2824.40	7850.30
	700	2	0	8	0.36	0.44	0.02	7849.30	2983.00	5620.20
	750	0	4	6	0.50	0.18	0.05	7968.70	4544.00	8543.20
	Ave.	1.17	2.67	6.17	0.44	0.28	0.07	4751.32	2692.17	5523.43
$U=8$	500	4	2	4	0.23	0.26	0.08	2281.30	995.40	2999.60
	550	3	1	6	0.21	0.36	0.09	2944.30	2122.10	2373.30
	600	4	2	4	0.41	0.40	0.20	4465.70	2581.10	5870.20
	650	2	5	3	0.29	0.12	0.17	5703.00	3052.30	7886.70
	700	2	2	6	0.46	0.38	0.15	7662.70	3096.70	5816.90
	750	3	3	4	0.42	0.32	0.19	9077.90	4671.60	11135.10
	Average	3.00	2.50	4.50	0.34	0.31	0.15	5355.82	2753.20	6013.63
Overall Ave		1.69	3.50	4.81	0.47	0.20	0.12	2586.77	1669.29	3111.14

III.5. Summary and Conclusions

In this chapter, we considered ONDP which is a network design problem with explicit consideration of consolidating smaller loads into TL shipments. Our setting addresses the common practice seen in the small package and mail delivery industries where the commodities (loads) are refer to the shipment requirements between pairs of local centers. A commodity is transported between the local centers (its origin and destination) either directly or it follows a route that visits two regional centers and is transferred in a TL shipment after consolidation with other commodities between the regional centers. Observing this practice, we provided a compact mathematical formulation for the problem of the least cost commodity flow with TL consolidation considerations addressed explicitly. We observed that a special case of the model resembles a capacitated location problem with single-sourcing, which is itself an NP-hard problem, and the instances that we are interested in translate into very large scale and challenging versions of this problem since they imply a large number of customers (commodities) and an even larger number of potential locations.

In order to develop heuristic solution algorithms, we first proposed four distinct compound neighborhood structures. A compound neighborhood involves two main components, level-change and content-change, and the latter component combines various simple neighborhood functions in a certain way which may also involve a local search itself for the purpose of generating a neighboring solution. Given the complexity of the kind of *efficient solution representation* required in a heuristic framework, our compound neighborhood functions give us the opportunity to search the solution space efficiently using a branching strategy that relies on the level-change component, which is the common component in all four compound neighborhoods. Furthermore, the components provide a means for instilling intensification (via content-change) and diversification (via level-change) characteristics into the heuristic search algorithms. We devised three different heuristic algorithms based on local search, simulated annealing and tabu search. In each algorithm, each of the four compound neighborhoods can be used, thus giving rise to twelve different approaches. Also, each algorithm utilizes the branching strategy in a certain way that promotes relative efficiency.

Our extensive computational study using four different data sets representing varying problem sizes revealed that the serial type compound neighborhood functions (SN and SLSN) perform better in terms of solution quality although they also require the longest solution time on average. On the other hand, the parallel type function with local search (PLSN) performs poorly in terms of both solution quality and time. However, the neighborhood function PN performs best in terms of solution time and very satisfactorily (most of the time close to the SN and SLSN performance) in solution quality. These results are generally consistent independent of the heuristic algorithm with which they are used. From the algorithmic perspective, we observed that our tabu search approach (TSCB) provides better solutions, especially for the large instances, but at the expense of solution time. On the other hand, simulated annealing based SABB performs very satisfactorily in terms of solution quality and is best in terms of solution time, especially for large instances. We also note that, although the local search based LSDB performs well with small size problems, its performance diminishes with larger sized instances where the solution space grows significantly and the meta-heuristic based approaches become more effective. Nevertheless, the branching strategy that we introduced appears to be very effective in all of the approaches, and it is flexible enough to be employed in searching the solution space in different ways.

CHAPTER IV

TACTICAL NETWORK DESIGN PROBLEM

In view of globalization, growing economies, and increased competition, it becomes important for LTL and intermodal transportation businesses to make tactical and strategic plans and achieve higher operational efficiency. As we discussed in the previous chapters, intermodal transportation models are comparatively new Crainic and Kim (2005), and the traditional hub-and-spoke network models may not be suitable for addressing the need of dedicated models for network design problems in the context of LTL and intermodal transportation problems. In ONDP, we assume availability of appropriate capacity and that the solution of ONDP can help the operations manager to make operational level decisions. Such an assumption can only be justified if appropriate capacity is planned at the tactical level. In this chapter, we investigate tactical capacity planning where the capacity can be obtained either by owned fleet or by contractual agreement from service providers. The cost of acquiring capacity on an emergency or expedited basis is much higher than the regular price; therefore, appropriate capacity planning can help reduce the cost of emergency capacity acquisition. Other tactical decisions may include planning for human resources and other equipment.

The formal problem description for TNDP can be given as follows. In TNDP, we are given a network with multi-commodities as described in ONDP. The planning horizon in TNDP is longer than an operational problem, but shorter than a strategic period. Depending on the application, the tactical planning horizon may correspond to a month or a quarter or even a year. Tactical decisions addressed by TNDP include

1. connections and capacities in terms of the number of *truckload trips* between the consolidation and deconsolidation centers

2. assignment of commodities to consolidation and deconsolidation centers and, in turn, to transfer links.

The costs in the system include collection costs, linehaul transfer costs and distribution costs. We assume that capacity installments on linehaul transfer links can be set in fixed increments of truckload capacity with associated incremental costs. Also, we assume that a commodity can follow only a single route from origin to destination.

The main objective is to estimate transportation capacity to ensure the availability of resources at the operational level. The resources are either owned or acquired on a temporary basis via a rental agreement. Generally, last-minute acquisition of capacity costs more and causes unnecessary delay. Tactical planning of transportation capacity helps eliminate the need for such arrangements and improves efficiency. Another benefit of tactical planning is that it facilitates negotiation for better price from third party providers such as transporters, fleet owners, and equipment rental companies.

We have already discussed an "auxiliary" network that provides an abstraction of the general physical network. The reader is referred to 8 of Chapter I for details of constructing an auxiliary network from a physical network. In the following, we refer to the auxiliary network in Figure 4 to describe TNDP specific details. The auxiliary network for TNDP differs from the one for ONDP; for one thing, direct shipment is not considered in the TNDP, and therefore, the auxiliary network for TNDP does not include the set of arcs representing direct shipments; second, the arrows from consolidation centers to deconsolidation centers represent consolidations that may consist of several TLs as opposed to the individual TLs in ONDP. In fact, they represent truckload trips over a time period equivalent to a tactical planning horizon. In the example given in Figure 4 in Chapter I, each of the arrows $2 \rightarrow 10$

and $5 \rightarrow 8$ represent a capacity in terms of number of truckload trips spanned over the tactical planning period.

Under these operational and configurational characteristics, our objective is to determine the linehaul links between the regional centers (consolidation and deconsolidation centers), the number of truckload trips assigned to each linehaul link, and the routing and consolidation of commodities so that the total cost of transportation is minimized. The total cost has three components: collection, distribution, and linehaul transfer from consolidation to deconsolidation centers.

The remainder of this chapter is organized as follows. Next, in Section IV.1, we develop an integer program formulation for our problem. This is followed by Section IV.2, in which we describe the basic ingredients of heuristic solution approach in detail, including our compound neighborhood functions. In Section IV.3, we present Lagrangian relaxation based heuristic to find lower bounds which utilizes upper bound as obtained in section IV.2 to update the Lagrangian multipliers in subgradient optimization framework. In Section IV.4, we present the results of our computational tests regarding the performance of the approaches and, in Section IV.5, we provide a summary of our conclusions and future research directions.

IV.1. The Model

Many different mathematical formulations can be used to model a problem. These formulations may differ from each other in types and number of variables, number of constraints, and several other characteristics such as LP bound. The mathematical formulation has bearing on the choice of solution method and, consequently, algorithm development. We developed several different models and compared them on such criteria as size of the formulation (number of variables and number of constraints), LP bound and branch-and-cut performance. We selected a pure integer model that overall performed best in our empirical study. In order to develop a mathematical formulation utilizing the auxiliary graph, we will use the notation given below.

Parameters:

w_i the amount of flow for commodity p_i

U capacity of truck

β cost of *truckload-trip* per mile

α^f LTL transportation cost per unit per mile between $\mathcal{F}$ and $\mathcal{J}$

α^t LTL transportation cost per unit per mile between $\mathcal{K}$ and $\mathcal{T}$

d^f_{ij} distance between f_{p_i} and consolidation center j

d^t_{ki} distance between deconsolidation center k and t_{p_i}

d_{jk} distance between centers j and k

Decision Variables:

z_{ijk} 1 if commodity i is assigned to a transfer link (j,k), 0 o.w.

y_{jk} number of *truckload-trips* installed on the link (j,k)

Then, the problem can be formulated as follows:

Objective and Constraints:

$$\text{Min} \quad \sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} w_i(\alpha^f\, d^f_{ij} + \alpha^t\, d^t_{ki})\, z_{ijk} + \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} d_{jk}\beta\, y_{jk} \tag{4.1}$$

subject to

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} z_{ijk} = \quad 1, \qquad \forall i. \tag{4.2}$$

$$\sum_{i\in\mathcal{P}} w_i z_{ijk} \leq U\, y_{jk}, \qquad \forall j,k. \tag{4.3}$$

$$y_{jk} \in \mathcal{Z}^+, \text{ and } z_{ijk} \in \{0,1\}, \qquad \forall i,j,k. \tag{4.4}$$

In the objective function given by expression (4.1), the first term represents the total transportation cost for the collection and distribution operations in graphs $G^C(\mathcal{F} \cup \mathcal{J}, A_{FJ})$ and $G^D(\mathcal{K} \cup \mathcal{T}, A_{KT})$, respectively; the second term represents the total transportation costs for the linehaul transfers using TL shipments in $G^L(\mathcal{J} \cup \mathcal{K}, A_{JK})$. Constraint set (4.2) ensures that each commodity is shipped via exactly one consolidated shipment over a link. Constraint set (4.3) ensures that, for each link (j,k), the total weight of the commodities assigned does not exceed the capacity installed on that link. Constraint set (4.4) imposes standard binary restrictions on the decision variables.

IV.1.1. Relation to SSCFLP with Modular Capacity

TNDP is related to an extension of the single-source capacitated facility location problem (SSCFLP) which itself is an extension of CFLP with the additional requirement that each customer must be assigned to exactly one facility. Below, we describe how our problem is related to SSCFLP with modular capacity.

In SSCFLP with modular capacity, for each potential facility location ,there is a finite and discrete set (Modules) of allowable capacities and the objective is to choose subset of facilities to satisfy the demand at minimum cost. Staircase capacity is a special case of modular capacity in which all modules are of the same size.

A solution to our problem consists of commodities being assigned to linehaul transfer links. One can view a linehaul transfer link as a potential facility on which capacity can in installed in the form of truckload-trips and the commodities as the customers. Then, clearly, our problem can be modelled as a CFLP. Since our problem requires each commodity to be assigned to exactly one linehaul link, our problem is equivalent to single source CFLP or simply SSCFLP. Furthermore, since our problem assumes fleet of identical trucks, our problem is equivalent to SSCFLP with staircase capacity.

As we discussed in Chapter II, facility location problems, even in their simplest form, belong to the class of NP-Hard problems. Meaning that there does not exist a polynomial time algorithm for the problem. Moreover, in contrast to a general location problems, our problems are much larger, because number of customers in a location problems is of the order of nodes in a network, whereas number of commodities in our problem are of the order of square of number of nodes.

To the best of our knowledge, SSCFLP with staircase costs has not been solved in the literature, and there is only one published paper on SSCFLP with modular capacity by Correia and Captivo (2006). Correia and Captivo (2006) study the SSCFLP with modular capacity with some additional constraint. They suggest a Lagrangian heuristic utilizing tabu search for upper bound and a Lagrangian relaxation based lower bound in which they further relax integrality restriction on the assignment variables. In order to obtain the lower bound, they use method developed by Cortinhal and Captivo (2003). Cortinhal and Captivo (2003) consider modular capacitated location problem in which they obtain lower bound by Lagrangian relaxation dualizing assignment constraint. They further tighten the Lagrangian relaxation by adding a valid inequality. Their relaxation sub-problems break down into one problem for every facility. In order to solve sub-problem for a facility, they formulate linear program for each size in allowable modules for that particular facility, which is continuous relaxation of constrained knapsack and can be solved without resorting to simplex method. Finally, the capacity on each facility is selected based on the objective value of the linear program solved above and some additional valid inequality on lower and upper bounds on number of allowed number of facilities.

Our lower bound solution approach also relaxes the assignment constraint, however, our approach differs from theirs in that: we do not break the problem for each facility, instead we use additional surrogate constraints to tighten the relaxation. In our model we exploit the fact that each module is of the same size by utilizing an integer decision variable to model capacity as opposed to their binary variable for each module for each facility. This results in a smaller problem size of our model, and facilitates development of tighter relaxation by adding surrogate constraint which is not possible if we break the problem into sub-problem for each facility. In contrast to their work, our problem sizes are much larger because the number of customers in a location problem are the number of commodities in our problem which is a difference of the order of N^2 because every node-pair on the network can be defined as a commodity in the TNDP.

For relatively smaller instances, the formulation can be solved to optimal efficiently using commercial grade software such as CPELX; however, the computational time and memory requirements become quite prohibitive for large problem instances. Since our interest is in solving large instances, in this chapter, we suggest a Lagrangian Heuristic (LH) based method. In the Lagrangian Heuristics framework, we develop an upper bound heuristic using a compound neighborhood search method, and a Lagrangian relaxation-based lower bound. In our Lagrangian heuristic, the lower bound solution prescribed by Lagrangian relaxation is utilized by our compound neighborhood search method to obtain a better upper bound, and the upper bound thus obtained is utilized to update the Lagrangian multipliers in the sub-gradient optimization. The efficiency of our Lagrangian framework is demonstrated by the computational study presented at the end of the chapter.

IV.2. Upper Bound Heuristic

In this section, we first describe the solution representation and objective function evaluation method which is frequently used during the heuristic solution process. Later, we provide construction heuristic to generate initial feasible solutions as inputs to the solution improvement (heuristic search) algorithm. Also in this section, we present the details of our compound neighborhood functions which are important ingredients in the metaheuristic simulated annealing to be described later.

IV.2.1. Solution Representation and Objective Function Evaluation

In our problem, a solution represents a set of commodities partitioned into disjoint and mutually exclusive sets of commodities. All commodities in a particular disjoint set are shipped over the same linehaul transfer link. Note that, if the cumulative load of commodities in a disjoint set is more than one truckload, then the commodities may be shipped in different trucks installed on that linehaul transfer link. We use S_l to represent a disjoint set of commodities assigned to the same linehaul transfer link and $\mathcal{S}$ to denote the set of all such disjoint subsets. The objective function value for a solution $\mathcal{S}$ is the total transportation cost implied by it. The total transportation cost can either be obtained by solving a small integer program or be approximated by using a heuristic. The heuristic is computationally inexpensive while an integer program may take longer to solve. We denote the exact objective value function by $Z^*(\mathcal{S})$ and the approximate objective value by $\tilde{Z}(\mathcal{S})$.

IV.2.1.1. Exact Objective Function Value

To model the integer program to evaluate the transportation costs implied by a solution $\mathcal{S} = \{S_1 \ldots S_L\}$, we use notation similar to the notation used in the model formulation.

Decision Variables:

z_{ljk} 1 if commodity set $S_l \in \mathcal{S}$ is assigned to a transfer link (j,k), 0 o.w.

y_{jk} number of *truckload-trips* installed on the link (j,k)

Then, the objective function value associated with $\mathcal{S}$ is given by :

$$Z^*(\mathcal{S}) = \text{Min} \quad \sum_{S_l \in \mathcal{S}} \sum_{i \in S_l} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} w_i(\alpha^f\, d_{ij}^f + \alpha^t\, d_{ki}^t)\, z_{ljk} + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} d_{jk}\beta\, y_{jk} \tag{4.5}$$

subject to

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} z_{ljk} = 1, \qquad \forall S_l \in \mathcal{S} \tag{4.6}$$

$$\sum_{S_l \in \mathcal{S}} \sum_{i \in S_l} w_i z_{ljk} \le U\, y_{jk}, \qquad \forall j,k. \tag{4.7}$$

$$y_{jk} \in \mathcal{Z}^+, \quad \text{and } z_{ljk} \in \{0,1\}, \qquad \forall l,j,k. \tag{4.8}$$

In the integer program shown above, the first terms in the objective function give by (4.5) represents total transportation cost of collection and distribution and the second term represents the linehaul transfer cost. Constraint (4.6) makes sure that each commodity set is assigned to exactly one link, and constraint (4.7) ensures that every link has sufficient capacity to haul the commodity sets assigned to it. Constraint (4.8) represents integrality restrictions on variables $\mathbf{y}$ and binary restrictions on variables $\mathbf{z}$.

If we replace $i \in \mathcal{P}$ by $S_l \in \mathcal{S}$, our original problem formulation (4.1)-(4.4) is identical to the objective function evaluation formulation above , however, the size of the IP to be solved for the same problem is much smaller in the objective function evaluation formulation because $|\mathcal{S}|$ is much smaller than $|\mathcal{P}|$. The approximate objective function value for a solution $\mathcal{S}$ can be calculated by a simple heuristic that we describe next. For each disjoint set $S_l \in \mathcal{S}$, we determine the transfer link (j, k), where $j \in \mathcal{J}$ and $k \in \mathcal{K}$, that provides the lowest collection-transfer-distribution cost via enumeration over the $|\mathcal{J}| \times |\mathcal{K}|$ transfer links. Once all the disjoint set $S_l \in \mathcal{S}$ are assigned to their nearest links, the implied capacity to be installed on a particular link (j, k) is calculated as the minimum number of truckload-trips required to haul the cumulative load assigned to that link. The objective function evaluation procedure described here provides a computationally inexpensive method for assigning a value depicting the goodness of the solution. In several cases, it may be very close to the exact value $Z^*(\mathcal{S})$.

The solution $\mathcal{S}$ consists of partitions $S_l \in \mathcal{S}$ which are assigned to various links in the process of the objective function evaluation. It may happen that more than one partition, say S_1, S_2 and S_3 get assigned to the same link. In such a case, we define a new partition S' which consists of all the commodities in S_2, S_2 and S_3 and redefine the solution as $\mathcal{S} = \mathcal{S} \cup \{S'\} \setminus (\{S_1\} \cup \{S_2\} \cup \{S_3\})$. Clearly, the objective function evaluation then may result in a solution with a reduced level of consolidation.

Since the objective function evaluation is used several times in our neighborhood search routines, we use $\tilde{Z}(\mathcal{S})$ for comparing the neighborhood solutions with each other. However, we update the objective value assigned to an incumbent solution by $Z^*(\mathcal{S})$. The use of approximate objective function evaluation enables us to search the neighborhood quickly. For clarity, although both $\tilde{Z}(\mathcal{S})$ and $Z^*(\mathcal{S})$ have an objective function value that corresponds to a solution $\mathcal{S}$, their purposes are different. The heuristic objective function evaluation is used to assign a number to the neighborhood solution for comparison purposes, and the main criterion is to be reasonable for comparison as well as computationally inexpensive because a large number of neighborhood solutions need to be compared. The exact objective function value is used to update the incumbent solution's value in the local search procedure explained later.

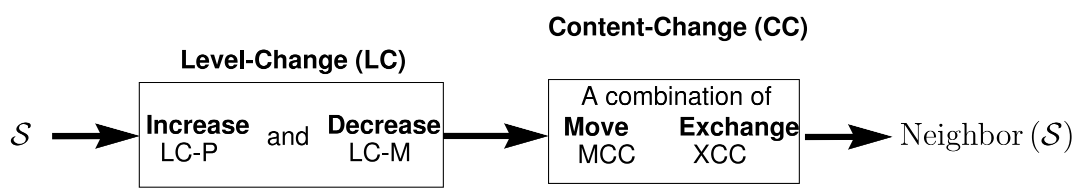
IV.2.2. Construction Heuristics

Based on the solution representation described above, the construction of an initial feasible solution clearly consists of partitioning commodities. There are a number of possible ways to obtain such partitions. We conducted empirical tests on several ideas based on random selection, greedy selection and a combination of the two. We found the greedy heuristic to perform better; therefore, below we present the initial solution construction method which is greedy in nature. In the construction method, for each commodity $i \in \mathcal{P}$, we select the transfer link (j, k) which has the lowest score defined by $\delta_{ijk} = w_i\,(\alpha^f\, d^f_{ij} + \alpha^t\, d^t_{ik} + \frac{\beta}{U} d_{jk})$. Notice that the score δ_{ijk} is a measure of the total cost of transferring commodity i from its origin to its destination via link (j, k) assuming 100% capacity utilization. Performing the above described score based link selection for each commodity partitions the commodities into sets (may include singleton sets) of commodities assigned to the same link. A collection of the thus created partitions is our initial solution $\mathcal{S}$.

IV.2.3. Components of Compound Neighborhoods

We developed a compound neighborhood search method in the previous chapter for ONDP. As we will show shortly, we can use the same framework here as well as the modified definition of the ingredient functions that adapt to the problem structure. Just as in ONDP, we begin by observing that in any given solution $\mathcal{S}$, we can identify two key attributes including the consolidation level defined as the number of partitions $|\mathcal{S}|$, and the composition of partitions $S_l \in \mathcal{S}$. Given a solution, a neighborhood function modifies these key attributes in order to generate neighboring solutions in a heuristic search framework. Since the neighborhood functions that we develop for this purpose utilize simple operations in various combinations for modifying the key attributes, we call them compound neighborhood functions. There are two essential components of a compound neighborhood function: the *Level-Change* (LC) and the *Content-Change* (CC). The LC component perturbs the consolidation level $|\mathcal{S}|$ in a solution $\mathcal{S}$, and the CC component modifies the contents of partition $S_l \in \mathcal{S}$. We define a compound neighbor of a given solution $\mathcal{S}$ as a solution obtained by first applying an operation in the LC component followed by a combination of operations in the CC component. In the latter, a specific combination is called a CC method. These components and their operations are outlined in Figure 10.

Figure 10 Components of Compound Neighborhoods



The **LC component** is comprised of two operations which are abbreviated as LC-P() and LC-M(). Given a solution $\mathcal{S}$, LC-P($\mathcal{S}$) gives a new solution with a consolidation level $|\mathcal{S}|+1$ whereas LC-M($\mathcal{S}$) gives a new solution with a consolidation level $|\mathcal{S}|-1$. Note that the number of possible consolidations in the system varies from a minimum of 1 consolidated shipment to a maximum M^2 consolidated shipments, one on every link. Both of the operations LC-P() and LC-M modify the level within this range. In operation LC-P(), to increase the consolidation level by one, we create a new empty set of commodities and repeatedly select a commodity at random from existing sets one at a time to assign it to the new set until the new set has a total of $\frac{N}{|\mathcal{S}|+1}$ commodities.

Similarly, in operation LC-M(), we reduce the consolidation level of a solution $\mathcal{S}$ by one consolidation by disaggregating one of the partitions and distributing its content to other partitions. We disaggregate the consolidation which has a load that leads to the most undesirable capacity utilization. Let $LOAD(S_i)$ denote the total weight of the commodities in the consolidated shipment $S_i \in \mathcal{S}$, then unutilized capacity is given by $U-(LOAD(S_i)\%U)$, e.g. for $LOAD(S_i) = 25, U = 10$, the unutilized capacity is given by $10 - (25\%10) = 5$. In operation LC-M(), we disaggregate the set $S_i \in \mathcal{S}$ having the lowest utilization. We randomly assign the disaggregated commodities to the remaining sets one at a time. Addition of a new commodity may cause the consolidated shipment to require an additional trucks. We try to avoid assigning the disaggregated commodities to such shipments. If it is impossible to assign the disaggregated commodities to a shipment without requiring an additional truck, we assign it to a randomly selected consolidation.

The **CC component** modifies the composition of sets $S_l \in \mathcal{S}$ using local search with simple move and exchange neighborhood functions. Recall from the solution method developed in the case of ONDP, move corresponds to moving a commodity from a set S_i to a set S_j (MCC) where $i \neq j$ and the exchange consists of a pair-exchange operation corresponding to exchanging a pair of commodities between a set S_i and another set S_j (XCC) where $i \neq j$. We notice that in TNDP, there are only two content change methods possible MCC and XCC as opposed to five in ONDP, namely XCC, XCD, MCC, MCD and MDC. We found that a neighborhood spanned by just two functions as defined above is not sufficient to generate good solutions. Therefore, in TNDP we modify the procedures MCC and XCC to pursue intensified searches by moving or exchanging *two* commodities instead of just one commodity as in the ONDP. Below we describe the XCC and MCC operations specific to TNDP.

In XCC, given partitions $S_1, S_2 \in \mathcal{S}$ involved in the exchange, we identify two links $(j,k)^1$ and $(j,k)^2$ that have the smallest total cost of collection, transfer and distribution with respect to partitions S_1 and S_2, respectively. Then, we calculate the cost of collection and distribution for all commodities in set $\mathcal{S}_1$ as if they were assigned to the link $(j,k)^2$ and select the two commodities with the lowest cost. Similarly, we identify two commodities in set S_2 that have the minimum cost of collection and distribution as if they were assigned to the link $(j,k)^1$. The XCC returns the new solution obtained by exchanging the two commodities both from partitions S_1, S_2. The solution thus obtained is accepted if the corresponding objective function value is better than the original solution before the exchange.

The MCC operation gives a solution obtained by moving 2 commodities from one partition to another partition. In MCC, we identify the least cost link (j,k) corresponding to the commodity set at the receiving end of the move operation. Similarly to XCC, in order to identify two commodities to be moved from the providing set, we calculate the cost of collection and transfer for all the commodities in the providing set assuming the commodities are assigned link (j,k), and select the two commodities with the lowest cost. The resulting solution is accepted if the corresponding objective function value is better than the original solution.

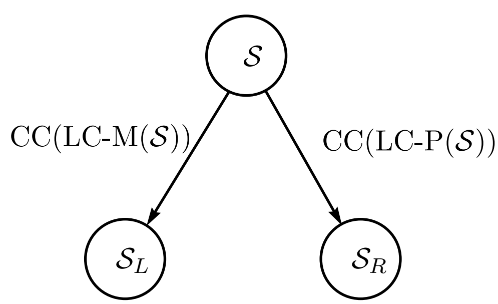
These simple neighborhood functions can be combined to prescribe CC methods that generate a neighboring solution of a given solution provided by the LC operation. We combine these methods in a serial fashion such that the first improving solution in XCC provides the initial solution for MCC. An important distinction of the compound neighborhoods described in ONDP is that here we do not use different compounding sequence CC-P and CC-M. Instead, the CC is identical for the solutions obtained from both LC-P and LC-M.

Notice that, just like in ONDP, the use of the LC and CC components in this fashion facilitates the incorporation of two desired characteristics in any heuristic search procedure. The LC component promotes *diversification* during the search of the feasible solution space. On the other hand, the methods of the CC component provide the opportunity for *intensification* in a solution subspace via the combined use of simple neighborhood functions.

IV.2.4. Generic Notation and Branching

We define the generic notation that we used in the previous chapter. Again, the function ConstructionHeuristic() refers to obtaining an initial solution which can be performed by applying procedure explained in Section IV.2.2. In these local search procedures, we again take the first improving solution at each iteration. We define branching on a node representing a current solution $\mathcal{S}$ as shown in Figure 11. The left child of node $\mathcal{S}$, denoted by $\mathcal{S}_L$, is a solution obtained by LC-M() followed by CC() and the right child of node $\mathcal{S}$, denoted by $\mathcal{S}_R$, represents a solution obtained by LC-P() followed by CC().

Figure 11 Branching on a Solution $\mathcal{S}$ in TNDP



The compound neighborhood search framework that we use in TNDP is similar to the one used in ONDP in the sense that both use two key attributes to define a neighborhood, and both level change and content change neighborhoods are based on similar type of neighborhood search ideas. Level change is still based on increasing and decreasing the level, and content change neighborhoods still use exchanges and move based neighborhood search. The key differences are in the implementation of the ingredient functions as summarized below.

1. In ONDP, the search tree has two levels, while in TNDP the search tree has only one level.

2. In ONDP, there are 3 move based and two exchange based neighborhoods while in TNDP we have one move and one exchange neighborhood.

3. In ONDP, we tried four methods of compounding the neighborhood searches, the PN, PLSN, and SN and SLSN, and three metaheursitics LSDB, SABB and TSCB. Based on computational results in ONDP, we have confined TNDP to only one method, SN, for compounding the search and SABB for the metaheuristic.

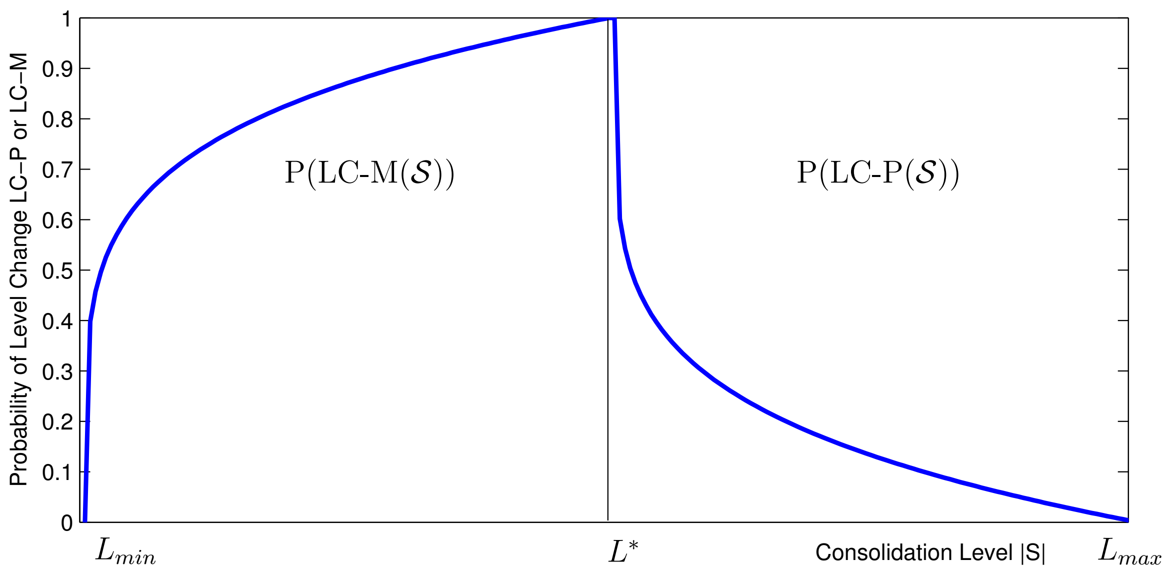
4. Instead of pair-wise exchanges and single moves, TNDP uses two-exchanges and two-moves and a more sophisticated way of selecting candidate commodities for the moves and exchanges.

IV.2.5. Simulated Annealing with Biased Branching (SABB)

The branching strategy that we introduced in ONDP, and again adapted for TNDP as described above, helps reduce the number of necessary neighborhood searches by biasing the search direction toward potentially improving good solutions only.

An overview of basic Simulated Annealing was presented in the previous chapter. In ONDP, our simulated annealing algorithm utilizes the branching on the solutions obtained during the search. We suggested a probabilistic branching strategy that introduced bias into the search in such a way that the iterates tend to move to the region of the solution space that is more likely to include good solutions. In general, the SABB algorithm presented here is a repetition of what we already presented in ONDP, except for changes in the parameter values and the use of a completely different bias branching strategy. This solution generation, given in lines 6-20 in Display 7, uses a biased branching strategy that is particularly specific in our case as described in detail below. If $\mathcal{S}^n$ improves upon the best solution ($\mathcal{S}^b$) to date, we update the best and current ($\mathcal{S}^c$) solutions and start a new iteration of Metropolis. On the other hand, if $\mathcal{S}^n$ is non-improving, we accept it as the current solution with a probability $e^{-\Delta/\mathtt{T}}$ where Δ is the absolute difference between the current and new solution and $\mathtt{T}$ is an algorithm parameter known as temperature. This mechanism provides an opportunity for accepting the uphill moves mentioned above. The parameter $\mathtt{T}$ is usually high for initial Metropolis runs so the acceptance probabilities are high and diversification in the search is promoted. After each Metropolis run, the temperature is decreased before the next one starts, thus providing an overall decreasing sequence of temperatures, usually in a geometric fashion. This is achieved using a factor γ (typically a value less than and close to one), i.e., $\mathtt{T}$ is updated as $\gamma\,\mathtt{T}$. Each Metropolis procedure is executed at a fixed temperature for a certain number of iterations $\mathtt{M}$ which is another algorithm parameter. Similar to $\mathtt{T}$, we also update the parameter $\mathtt{M}$ after each Metropolis run using another factor ϕ, i.e., $\mathtt{M}$ is updated with $\phi\,\mathtt{M}$; however, in this case, we choose a factor value that is greater than one. A cooling schedule set with these general characteristics promotes intensification in the search as the overall algorithm proceeds while encouraging diversification to reach regions with good solutions in the initial stages. The overall SABB Algorithm is terminated when the required number of iterations in a Metropolis loop exceeds a preset algorithm parameter value MAX_M. The complete SABB algorithm is given in Display 7. Note that the algorithm parameter values specified in the initialization step in Display 7 were obtained after some fine-tuning and used in our computational tests presented in Section IV.4.

Figure 12 Biased Probability in Simulated Annealing



Similarly to SABB in ONDP, here also we use a biased branching strategy, but with certain adaptations. This biased branching helps avoid searching in directions that are less likely to produce better solutions. Note that in TNDP, the level of consolidation varies from a minimum of $L^{min} = 1$ consolidations to a maximum of L^{max} consolidations where L^{max} is given by $\min\{N, M^2\}$. Let us denote the desirable level of consolidation under perfect consolidation by $L^* = \min\{\lceil W(\mathcal{P})/U \rceil, M^2\}$ and the consolidation level of the current solution by L. Then the bias in the branching direction is decided by a ratio r calculated as:

$$r = \begin{cases} \frac{L-L^*}{L^{max}-L^*} & \text{if } L > L^* \\ \frac{L^*-L}{L^*-1} & \text{if } L \leq L^* \end{cases} \tag{4.9}$$

The branch $\mathcal{S}_{\mathcal{L}}$ for the case $L \geq L^*$ is selected with a probability ($P = r^a$) and branch $\mathcal{S}_{\mathcal{R}}$ for the case $L \leq L^*$ with a probability ($P = r^a$) where r is calculated as given by 4.9. The parameter a can be selected empirically, and we found $a = 0.2$ to work well. The probability distribution function plotted for a test problem is displayed in Figure 12. We implement this branching (see Figure 11) strategy in lines 6-20 of the SABB Algorithm given in Display 7.

IV.3. Lower Bound

It is known that even moderate sizes network design problems are often extremely difficult to solve using exact methods; therefore it is difficult to obtain benchmark solutions to evaluate the quality of the feasible solutions obtained by neighborhood search method. The most popular approach for obtaining lower bounds is by relaxation. In relaxation, the idea is to search the relaxed (larger) solution space, and minimize an objective function of smaller or equal value (Wolsey, 1998). Linear programming relaxation, combinatorial relaxation and Lagrangian relaxation are some of the important approaches found in the literature. Solutions to relaxed problems can also be used to develop algorithms for finding feasible solutions, e.g. LP based branch and cut approach, Lagrangian based dual ascent approach. Below, we evaluate two methods, LP relaxation and Lagrangian relaxation for developing lower bounds for the problem.

Display 7 Simulated Annealing Algorithm **SABB**()

initialize M=15, MAX_M=55,
T=500, $\gamma = 0.9$, $\phi = 1.2$
$\mathcal{S}^c$ = ConstructionHeuristic(), $\mathcal{S}^b = \mathcal{S}^c$
while M $\leq$ MAX_M **do**
  Set M$'$ = M
  repeat {*Metropolis Loop*}
    if $L > L^*$ **then**
      Calculate $r = \frac{L-L^*}{L^{max}-L^*}$
      if $r^a >$ rand[0,1] **then**
        $\mathcal{S}^n = \mathcal{S}^c_L$
      else
        $\mathcal{S}^n = \mathcal{S}^c_R$
      end if
    else
      Calculate $r = \frac{L^*-L}{L^*-1}$
      if $r^a >$ rand[0,1] **then**
        $\mathcal{S}^n = \mathcal{S}^c_R$
      else
        $\mathcal{S}^n = \mathcal{S}^c_L$
      end if
    end if
    $\triangle = Z^*(\mathcal{S}^n) - Z^*(\mathcal{S}^c)$
    if $\Delta < 0$ **then**
      $\mathcal{S}^c = \mathcal{S}^n$
      if $Z^*(\mathcal{S}^c) < Z^*(\mathcal{S}^b)$ **then**
        $\mathcal{S}^b = \mathcal{S}^c$
      end if
    else
      if rand[0,1] $< e^{-\Delta/T}$ **then**
        $\mathcal{S}^c = \mathcal{S}^n$
      end if
    end if
    M$'$ = M$'$ - 1
  until M$' = 0$
  T = γT; M = $\lfloor \phi$M$\rfloor$
end while
RETURN $Z^*(\mathcal{S}^b)$ and $\mathcal{S}^b$

Linear programming (LP) relaxation of the original problem is one of the easiest ways to obtain lower bounds. In most cases the LP bound is weak, but it can be improved by adding valid inequalities or cuts. Disaggregation constraints are valid inequalities in the form of $z_{ijk} \leq y_{jk} \quad \forall j, k$, which, though they are redundant for the integer program, tighten the LP bound. For example, consider the LP polyhedra defined by $LP = \{z_i : .8\,z_1 + .7\,z_2 \leq 2\,y, 0 \leq z \leq 1, y \geq 0\}$; the polyhedra including disaggregation constraints is $LP' = \{z_i : .8\,z_1 + .7\,z_2 \leq 2\,y, z_1 \leq y, z_2 \leq y, 0 \leq z \leq 1, y \geq 0\}$. It is obvious that $LP' \subseteq LP$ because $\{z_1 = 1, z_2 = 1, y = .8\} \in LP$ but $\notin LP'$. Our empirical study verified that the disaggregation constraints define a tighter LP polyhedra and help provide a better LP bound. Valid inequalities in the form of disaggregation constraints provides tighter LP bounds which often helps branch and cut convergence, but at the cost of having to solve a *larger* LP at each branch and bound tree node. Our empirical tests show that although adding valid inequalities obtained from disaggregation constraints resulted in faster convergence in the beginning, it suffers from tailing off issues. For larger problems, the impact was even more severe. Therefore, the LP relaxation based approach is not suitable for large instances of our problems.

The Lagrangian relaxation approach has been successfully applied to many difficult problems in distribution network design, facility location, and other areas. The Lagrangian relaxation-based approach involves relaxing a set of constraints and introducing them in the objective function with a multiplier vector called a Lagrangian multiplier vector. The relaxed problem is called a Lagrangian relaxation subproblem and the solution to this relaxed problem provides a lower bound (Z_{LB}) on the optimal solution of the main problem. One way to obtain Lagrangian relaxation is to dualize the capacity constraint (4.3) which results in Lagrangian subproblems consisting of constraints set (4.2) and standard nonnegativity and integrality restrictions.

This relaxation is separable in each commodity $i \in \mathcal{P}$ and is solvable by inspection. Unfortunately, this relaxation, although simpler to solve, suffers from the integrality property. A Lagrangian relaxation is said to have the *integrality property* if the solution to the Lagrangian subproblem is unchanged when the integrality restriction is removed. Furthermore, a Totally Unimodular constraint matrix defines an integral polyhedra, i.e., all extreme points of the linear program are integral. If a Lagrangian relaxation subproblem has the integrality property, then the Lagrangian bound can never be tighter than the LP bound. A matrix A is totally unimodular (TUM) if every square submatrix of A has a determinant of either 0, 1 or -1. In the case of this relaxation, the constraint matrix for each commodity is simply a 1 by M^2 matrix, i.e. row vector, and its elements are either 0 or 1. Clearly, the constraint matrix is totally unimodular, and therefore, the polyhedra prescribed by constraint set (4.2) is integral and the Lagrangian subproblem has the integrality property, implying that the lower bound provided by this relaxation can never be better than the linear programming relaxation.

If we add the valid inequalities ($z_{ijk} \leq y_{jk} \forall i, j, k$) to the Lagrangian subproblem in the previous relaxation, it can be shown that the Lagrangian subproblem polyhedra defined by constraints (4.10), (4.11) and (4.12) is exactly the uncapacitated facility location (UFL) polyhedra (shown below by 4.10 - 4.12) which is a known NP-Hard problem to solve.

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} z_{ijk} = 1 \qquad \forall i. \tag{4.10}$$

$$z_{ijk} \leq y_{jk} \qquad \forall j, k. \tag{4.11}$$

$$y_{jk} \in \mathcal{Z}^+, \ z_{ijk} \in \{0, 1\}, \qquad \forall i, j, k. \tag{4.12}$$

Since Lagrangian relaxation obtained by relaxing the capacity constraints yields poor bounds and an attempt to tighten the bounds by adding valid inequalities results in a hard problem to solve, we explore Lagrangian relaxation obtained by dualizing the demand constraint (4.2). The resulting Lagrangian relaxation is given by the formulation shown below.

$$\text{Min} \quad \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} (W_{ijk} - \mu_i) z_{ijk} + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} T_{jk}\, y_{jk} + \sum_{i \in \mathcal{P}} \mu_i \tag{4.13}$$

subject to

$$\sum_{i \in \mathcal{P}} w_i z_{ijk} \ \leq U\, y_{jk} \qquad \forall j, k. \tag{4.14}$$

$$\mu_i \geq 0, y_{jk} \in \mathcal{Z}^+, \text{ and } z_{ijk} \in \{0, 1\}, \qquad \forall i, j, k. \tag{4.15}$$

In the relaxation shown above, μ_i is the lagrangian multiplier corresponding to the constraint (4.2). Since the Lagrangian relaxation subproblem does not have integrality property, it is expected to provide better or at least equal lower-bounds than the linear programming relaxation of the original problem. The subproblem is of the binary knapsack type with the knapsack size itself being a decision variable which is a difficult problem. Since the subproblem needs to be solved at every iteration of the Lagrangian heuristic, it is important that we find some computationally inexpensive solution method. We devise a variable relaxation to obtain a faster lower bound as explained below.

Variable Relaxation: We assume single sourcing constraints, implying that a commodity must be assigned to exactly one linehaul link ($(j, k) - pair$). When we drop the single sourcing constraint by redefining the variable z_{ijk} as a real variable instead of a binary one, we call the resulting relaxed problem the *variable relaxation.*

The variable relaxation problem is easier to solve (i.e., results in smaller runtime), but it would obviously yield a lower bound as it is a relaxation of the original formulation. We confirm by the empirical test that variable relaxation results in an over 50% reduction in runtime with negligible compromise (less than 0.1%) in solution quality.

Furthermore, it is possible to tighten this lower bound by adding two surrogate constraints (4.18) and (4.19). The complete Lagrangian relaxation formulation including surrogate constraints and variable relaxation is give below.

$$Z_{LB}(\boldsymbol{\mu}) = \text{Min} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} (W_{ijk} - \mu_i) z_{ijk} + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} T_{jk}\, y_{jk} + \sum_{i \in \mathcal{P}} \mu_i \tag{4.16}$$

subject to

$$\sum_{i \in \mathcal{P}} w_i z_{ijk} \leq U\, y_{jk} \qquad \forall j, k. \tag{4.17}$$

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} U\, y_{jk}, \geq \sum_{i \in \mathcal{P}} w_i \tag{4.18}$$

$$\sum_{i \in \mathcal{P}} z_{ijk} \;\; \geq N \tag{4.19}$$

$$\mu_i \geq 0, y_{jk} \in \mathcal{Z}^+, \text{ and } 0 \leq z_{ijk} \leq 1 \qquad \forall i, j, k. \tag{4.20}$$

The surrogate constraint (4.18) dictates that the total capacity in terms of truckload trips installed over all links must be greater than, or at least equal to, the total demand. This surrogate constraint can be derived from constraint (4.3) by summing it over all links (j, k) and using the constraint (4.2). Similarly, constraint (4.19) implies a surrogate constraint obtained by summing the relaxed constraint (4.2) over all commodities $i \in \mathcal{P}$.

Also, notice that the Lagrangian relaxation subproblem (4.16)-(4.20) is an mixed integer program having only M^2 integer variables, which is easier to solve than the pure integer program with number of integer variables of the order of $NM^2 + M^2$.

IV.3.1. Lagrangian Heuristic

In general terms, a Lagrangian heuristic is an iterative procedure where at each iteration a lower bound $Z_D(\boldsymbol{\mu})$ using fixed μ_i values, an upper bound, using partial information from the lower bound solution, and updated lagrange multipliers are obtained. Iterations are continued until a stopping criterion is met. We use the following notation to describe the Lagrangian Heuristics for our problem.

Notations:

t	iteration number
T	maximum number of iterations
Z_{LB}^t, Z_{UB}^t	lower and upper bound at iteration t
$\underline{Z}, \overline{Z}$	best lower and upper bound
$\boldsymbol{\mu}^t$	vector of lagrange multipliers at iteration t
R^l	# of successive iterations with non-improving Z_{LB} (counter r^l)
R^u	# of cumulative iterations with less than ϵ% improvement (counter r^u)
λ	step size factor
d^n	step size to update lagrange multipliers at iteration v

The general framework for lagrangian heuristics is given in Display 8. The solution obtained from solving the Lagrangian relaxation can be converted into a feasible solution and used to generate an upper bound using the neighborhood search method. The information from the Lagrangian relaxation may be useful for speeding up the neighborhood search method by providing a good starting solution plus a better upper bound may further help the Lagrangian heuristic to converge faster.

Display 8 Lagrangian Heuristic Framework

Step 0: *Initialize:* t=1, $\overline{Z} = Z_{UB}^0 = \mathbf{SABB}(), \underline{Z} = Z_{LB}^0 = -\infty$,

$\mu^t, \alpha = 1.8.$ $r^l = r^u = 0.$

$\epsilon = 0.5$, $E = 100$

Step 1: $E = 100 \times \frac{Z_{UB} - Z_{LB}}{Z_{UB}}$

Solve $Z_{LB}(\boldsymbol{\mu}^t)$ to obtain a lower bound Z_{LB}^t.

If Z_{LB} satisfies constraints (4.2) set $Z_{UB} = Z_{LB}$, go to step 10.

Step 2: If $100 \times \frac{Z_{UB} - Z_{LB}}{Z_{UB}} - E \geq \epsilon$ Then $r^u \leftarrow r^u + 1$

If $Z_{LB}^t \geq \underline{Z}$, then $\underline{Z} = Z_{LB}^t$ and $r = 0$, otherwise $r^l \leftarrow r^l + 1$.

Step 3: Find a feasible solution as upper bound Z_{UB}^v.

Step 4: If $Z_{UB}^v \leq \overline{Z}$, then $\overline{Z} = Z_{UB}^v$.

Step 5: If $(\overline{Z} - \underline{Z})/\overline{Z} \leq \epsilon$, then go to step 10.

Step 6: If $r^l = R^l$, then $\alpha \leftarrow \alpha/2$

If $r^u = R^u$ Then $r^u = 1$ and $Z_{UB} = \mathbf{SABB}^*()$

Step 7: If termination criterion met then go to step 10.

Step 8: Calculate the step size for updating lagrange multipliers.

Step 9: Set $t \leftarrow t + 1$, Update the lagrange multipliers, go to step 1.

Step 10: Record the upper bound solution, stop.

A few remarks regarding the Lagrangian heuristic are in order. We use subgradient optimization to update the Lagrangian multipliers. In order to update the multiplier we first calculate the step size using the formula $d^t = \lambda^t \frac{UB-LB^t}{\sum_i(1-\sum_j\sum_k z_{ijk})^2}$, and then each multiplier is updated using the formula $\mu_i^{t+1} = max\{0, \mu_i^t + d^t(1 - \sum_j\sum_k z_{ijk})\}$. Notice that in step 6 of the Lagrangian heuristic as shown in Display 8, we update the Z_{UB} by using simulated annealing described in Display 7. However instead of starting the simulated annealing (refer to line 6 in Display 7) with an initial solution obtained from the ConstructionHeuristic(), we use the lower bound solution after making it feasible as per the procedure given below.

1. Define sets of commodities A, A^0, A^1 such that $A^1 = \{i : \sum_j\sum_k z_{ijk} = 1 \quad \forall i \in \mathcal{P}\}$, $A^0 = \{i : \sum_j\sum_k z_{ijk} = 0 \quad \forall i \in \mathcal{P}\}$ and $A = \mathcal{P} \setminus \{A^0 \cup A^1\}$

2. Retain the assignments for all commodities in set A^0

3. Assign every commodity i in set A^1 to the link $(j,k) = \arg\min\{\beta d_{jk} + w_i(d_{ij}^f + d_{ki}^t) \quad \forall j,k\}$

4. Assign every commodity in the set A to link $(j,k) = \arg\min\{\beta d_{jk} + w_i(d_{ij}^f + d_{ki}^t) \quad \text{for } j,k\}$ without increasing the number of TL requirements. If there is no such link, assign it to a randomly selected link.

5. Identify disjoint sets of commodities assigned to the same link and create feasible solution S. Assign an objective function value to $\mathcal{S}$ by $\tilde{Z}(\mathcal{S})$.

IV.4. Computational Study

In the preceding sections, we have described an integer programming formulation for our problem, simulated annealing based heuristic algorithms to find upper bound and a Lagrangian-based heuristic to find the lower bounds on the optimal solution to our problem. We also described a Lagrangian heuristics framework that uses SABB for the upper bound, Lagrangian relaxation for the lower bound, and a sub-gradient based approach for updating the Lagrangian multipliers. Our theoretical analysis suggests that the Lagrangian relaxation based lower bound is tighter than the lower bounds obtained by using LP relaxation and LP relaxation with disaggregated constraints. In this section, we present a computational test to verify our theoretical results and test the performance of the methods developed for lower and upper bounds.

IV.4.1. Experimental Setup

The process of generating test instances is presented on page 15.

Table 9 TNDP Experimental Problem Sets

Datasets	E	N_P	N	M	U
Benchmark Problems	15	15	$30, 40 \ldots, 100$	6	8
Dataset 1	25	25	$125, 150, \ldots, 200$	6	8
Dateset 2	40	40	$400, 450, 500$	6	8
Dataset 3	50	50	$800, 900, 1000$	6	8

The parameters to generate experiment data data are given in Table 9 which includes four data sets. In all of the data sets, A is set to 100, and the data sets differ from each other in terms of E, N_P, N, M and U. For each value of N, we randomly generate 10 instances. For simplicity, we choose $|\mathcal{J}| = |\mathcal{K}| = M$, which implies M^2 possible directed links for TL shipments. In all our problems instances, we chose $M = 6$ and $U = 8$ and, therefore, $\beta = 5$ per mile. We reduce the β used in TNDP to 5 instead of 6 in case ONDP to account for the price break that is possible due to load commitment to a third party. The first problem set includes 80 small instances where CPLEX provides benchmark results in the form of either an exact solution or upper and lower bounds upon termination with a runtime limit of 1800 seconds. Datasets 1 through 3 have comparatively larger numbers of commodities as given in Table 9.

IV.4.2. Computational Results

In section IV.3, we discussed various methods of generating lower bounds such as a LP Lower bound (LP), a lower bound generated by tighter LP that uses disaggregation constraints to tighten the formulation (LP-Tight), and a Lagrangian Bound (LR). Let us also denote by B&C the best lower bound reported by CPLEX at the time of termination. We showed the LP-Tight to be better than LP, the LR to be tighter than the LP, for the relaxation given by (4.16)-(4.20). Figure 13 shows lower bounds generated by all four methods LP, LP-Tight, LR and B&C. We observe that the lower bounds generated by LP and LP-Tight are the weakest. The LR and B&C lower bounds are tight for relatively smaller problems, however, as the problem size increases, the LR lower bound is higher than the B&C bounds.

We used CPLEX to obtain solutions for the benchmark problems instances with the stopping criterion of a 1800 seconds time limit. We benchmark the Lagrangian heuristic against the solution obtained from CPLEX. In both CPLEX and Lagrangian heuristics, upon termination we record the lower bound (LB) and upper bound (UB) and calculate the termination gap as $100 \times (Z_{UB} - Z_{LB})/Z_{LB}$. The summary of computational results for benchmark problems presented in Tables 10 and 11. In both the tables, column 2 to 5 show the CPLEX results and columns 6 to 9 show the Lagrangian heuristics. CPLEX is able to prescribe an optimal solution for all but one case for problems with values of N from 30 to 60. The average runtime for small instances is 17.2, which grows quickly to 603 seconds for $N = 60$. For the larger problems in the benchmark dataset, CPLEX is unable to prescribe an optimal solution within the set time limit of 1800 seconds. However, when we try to let CPLEX run for longer, it runs out of memory. The gaps are $7.42, 7.82, 6.70$ and 7.97 percent for values

Figure 13 Comparison of Lower Bounds

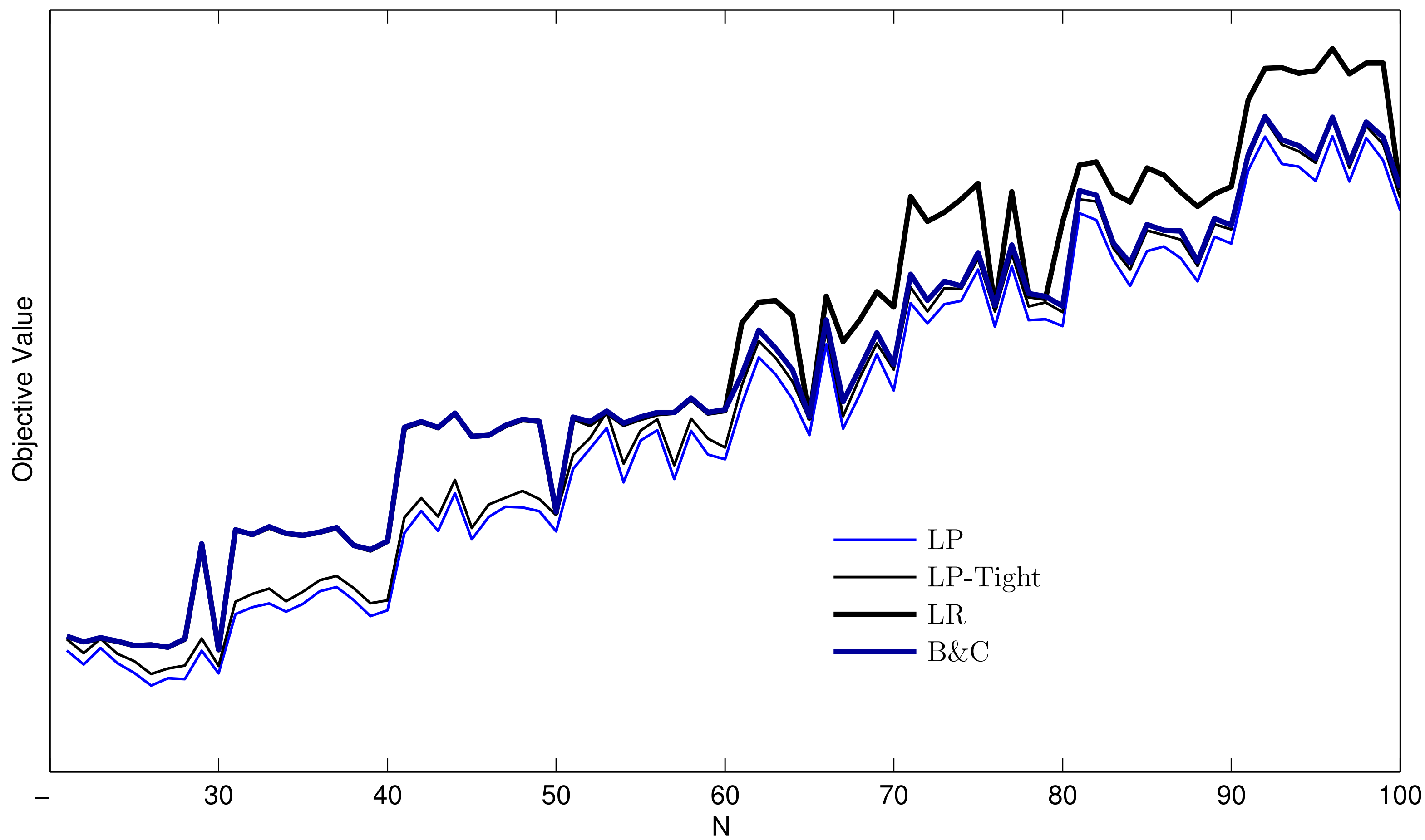


Table 10 Comparison Lagrangian Heuristic and CPLEX on Benchmark Dataset 0 ($N = 30 - 60$)

N	CPLEX				Lagrangian Heuristic			
	LB	UB	% Gap	Time	LB	UB	% Gap	Time
30	1122.27	1122.33	0.01	3	1121.12	1121.14	0.00	8
	1097.4	1097.5	0.01	13	1097.14	1097.24	0.01	13
	1116.16	1116.22	0.01	1	1113.67	1114.1	0.04	5
	1099.51	1099.54	0.00	6	1099.17	1099.27	0.01	4
	1079.49	1079.51	0.00	13	1079.46	1084.12	0.43	11
	1082.52	1082.61	0.01	13	1082.48	1089.32	0.63	7
	1072.38	1072.38	0.00	14	1071.98	1086.32	1.32	16
	1111.29	1111.29	0.00	18	1108.98	1109.19	0.02	8
	1548.03	1548.11	0.01	84	1546.95	1559.96	0.83	408
	1061	1061	0.00	7	1060.71	1060.73	0.00	8
Ave.			0.00	17.2			0.33	48.8
40	1611.85	1611.85	0.00	53	1611.64	1611.85	0.01	67
	1589.87	1589.94	0.00	166	1589.2	1600.54	0.71	75
	1626.11	1626.27	0.01	141	1624.16	1650.3	1.58	112
	1595.09	1595.17	0.01	115	1594.89	1610.52	0.97	54
	1586.7	1586.76	0.00	229	1585.89	1608.02	1.38	104
	1601.26	1601.39	0.01	69	1600.5	1600.51	0.00	30
	1622.17	1622.17	0.00	137	1621	1652.28	1.89	114
	1540.91	1540.99	0.01	36	1540.86	1540.82	0.00	10
	1520.65	1520.8	0.01	96	1518.84	1547.63	1.86	53
	1559.93	1559.93	0.00	74	1559.18	1559.41	0.01	48
Ave.			0.00	111.6			0.84	66.7
50	2083.22	2083.28	0.00	1181	2080.16	2131.65	2.42	101
	2108.89	2109.08	0.01	693	2107.04	2138.34	1.46	137
	2081.96	2082.14	0.01	596	2081.27	2109.09	1.32	96
	2148.49	2148.7	0.01	484	2146.7	2187.14	1.85	47
	2040.76	2040.89	0.01	525	2040.65	2057.18	0.80	101
	2046.24	2046.41	0.01	946	2045.76	2049.74	0.19	74
	2092.18	2092.38	0.01	721	2090.34	2126.67	1.71	83
	2119.25	2119.46	0.01	692	2117.01	2170.62	2.47	75
	2110.78	2110.99	0.01	632	2109.97	2139.87	1.40	76
	1694.24	1694.4	0.01	8	1690.08	1732.6	2.45	22
Ave.			0.01	647.8			1.61	81.2
60	2129.77	2129.98	0.01	271	2125.93	2183.94	2.66	23
	2109.57	2109.77	0.01	215	2095.06	2138.46	2.03	27
	2157.45	2157.67	0.01	52	2151.85	2295.86	6.27	30
	2102	2102.13	0.01	890	2096.55	2134.53	1.78	54
	2130.22	2130.43	0.01	23	2122.47	2176.25	2.47	8
	2151.5	2151.69	0.01	172	2144.74	2172.21	1.26	63
	2151.29	2151.5	0.01	970	2150.9	2199.03	2.19	81
	2217.4	2227.25	0.44	1801	2214.27	2272.59	2.57	63
	2150.9	2151.11	0.01	398	2147.71	2197.69	2.27	38
	2164.78	2165	0.01	1242	2159.55	2206.11	2.11	50
Ave.			0.05	603.4			2.56	43.7

of N as $70, 80, 90$ and 100, respectively and the runtime is 1800 seconds for most of the cases. Lagrangian heuristic performs quite well on all of the problems, and is able to find optimal solutions in several cases for $N = 30$ and 40. The average gap for N as 30 and 40 is below 1 percent with a runtime of 48.8 seconds and 66.7 seconds, respectively. For larger problems in the benchmark dataset, the Lagrangian heuristics prescribes solution within the range of a 2 to 3 percent gap which is much smaller than the 7 percent gap for CPLEX. The key feature of the Lagrangian heuristics is its small runtime in prescribing the good quality solutions. Even for large problems in the

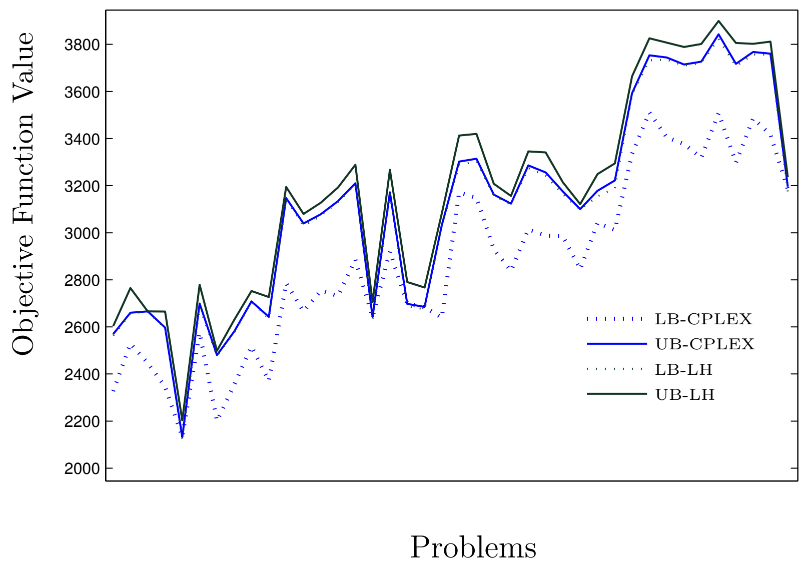
Table 11 Comparison of Lagrangian Heuristic and CPLEX on Benchmark Problems ($N = 70 - 100$)

N	CPLEX				Lagrangian Heuristic			
	LB	UB	% Gap	Time	LB	UB	% Gap	Time
	2327.12	2569.89	9.45	1801	2563.4	2603.68	1.55	150
	2529.48	2660.38	4.92	1801	2657.43	2765.18	3.90	65
	2446.24	2665.7	8.23	1801	2664.75	2665.9	0.04	72
	2346.33	2597.65	9.67	1801	2594.31	2665.56	2.67	312
70	2132.24	2132.45	0.01	58	2125.3	2203.58	3.55	81
	2576.1	2699.15	4.56	1801	2685.22	2779.5	3.39	45
	2200.78	2480.13	11.26	1801	2476.34	2498.48	0.89	413
	2355.86	2580.97	8.72	1801	2576.92	2630.47	2.04	107
	2517.56	2708.62	7.05	1801	2706.19	2752.25	1.67	51
	2370.94	2642.75	10.29	1801	2635.88	2727.26	3.35	279
Ave.			7.42	1626.7			2.31	157.5
	2786.08	3147.06	11.47	1801	3143.02	3194.73	1.62	116
	2666.07	3039.47	12.29	1801	3029.14	3080.07	1.65	406
	2753.54	3078.53	10.56	1801	3071.1	3127.53	1.80	270
	2732.01	3134.53	12.84	1801	3131.02	3192.4	1.92	247
80	2885.34	3210.16	10.12	1801	3203.63	3288.93	2.59	541
	2642.49	2642.75	0.01	207	2631.22	2706.71	2.79	49
	2920.36	3172.14	7.94	1801	3165.05	3267.22	3.13	81
	2697.44	2697.71	0.01	746	2686.29	2790.65	3.74	26
	2684.75	2684.94	0.01	738	2674.95	2767.23	3.33	112
	2639.22	3031.83	12.95	1800	3028	3081.15	1.73	350
Ave.			7.82	1429.7			2.43	219.8
	3170.59	3302.19	3.99	1801	3287.7	3412.76	3.66	63
	3148.82	3314.62	5.00	1801	3302.61	3419.96	3.43	86
	2930.11	3162.3	7.34	1800	3158.11	3208.16	1.56	61
	2838.45	3123.85	9.14	1801	3118.05	3156.34	1.21	145
90	3014.64	3286.01	8.26	1800	3274.29	3345.11	2.12	165
	2988.16	3256.46	8.24	1801	3242.22	3341.33	2.97	128
	2984.71	3175.18	6.00	1801	3163.04	3214.24	1.59	268
	2844.43	3100.43	8.26	1801	3097.71	3121.47	0.76	228
	3042.53	3178.32	4.27	1801	3154.98	3249.25	2.90	113
	3011.15	3222.17	6.55	1800	3188.95	3294.08	3.19	255
Ave.			6.70	1800.7			2.34	151.2
	3334.8	3592.41	7.17	1800	3586.17	3664.45	2.14	96
	3511.54	3753.61	6.45	1801	3732.22	3825.46	2.44	135
	3404.6	3744.02	9.07	1801	3735	3807.58	1.91	254
	3376.83	3714.5	9.09	1801	3709.81	3789.26	2.10	145
100	3318.59	3726.28	10.94	1800	3721.86	3801.8	2.10	224
	3507.8	3843.03	8.72	1801	3823.01	3899.48	1.96	253
	3297.94	3717.45	11.28	1801	3707.29	3805.68	2.59	655
	3485.62	3767.99	7.49	1800	3756.96	3802.7	1.20	181
	3416.4	3760.32	9.15	1800	3756.36	3811.32	1.44	261
	3184.44	3196.26	0.37	1801	3174.69	3235.75	1.89	29
Ave.			7.97	1800.6			1.98	223.3

benchmark dataset, the runtime for Lagrangian heuristic is quite small in comparison to the CPLEX.

We know that CPLEX uses a LP based branch and cut approach to generate lower bounds for the problem, and we showed that Lagrangian relaxation based lower bound is tighter than the LP bound for our problem. In Figure 14, we show the Upper and Lower Bounds generated by CPLEX and Lagrangian heuristic. We observe that, although in most cases, the CPLEX upper bound is very close to that generated by the compound neighborhood based upper bound, it is the tight lower bound generated

Figure 14 Comparison of Bounds: Lagrangian Heuristics Vs. Branch and Cut



by the Lagrangian that give the lagrangian heuristics an edge over the CPLEX's LP based branch and cut. On the basis of the encouraging computational results for benchmark problems, we can conclude that Lagrangian heuristic provides good quality lower and upper bounds.

Table 12 Lagrangian Heuristic Results for Dataset 1 ($N = 125 - 200$)

N	Lagrangian Heuristics			
	LB	UB	% Gap	Time
125	4706.74	4810.04	2.15	70
	4331.87	4461.28	2.90	107
	4405.31	4606.8	4.37	28
	4762.07	4996.88	4.70	192
	4351.04	4525.97	3.87	149
	4685.14	4809.85	2.59	32
	4539.28	4787.34	5.18	25
	4441	4544.89	2.29	140
	4551.67	4748.59	4.15	44
	4384.27	4491.99	2.40	432
150	5479.33	5628.02	2.64	271
	5503.22	5772.99	4.67	85
	5468.59	5952.83	8.13	76
	5586.64	5876.89	4.94	185
	5597.75	5789.09	3.31	156
	5637.01	5956.65	5.37	159
	5438.73	5932.62	8.32	161
	5614.94	5781.9	2.89	101
	5425.42	5591.91	2.98	163
	5411.01	5688.35	4.88	181
175	6080.98	6446.03	5.66	136
	6715.05	6905.42	2.76	98
	6132.12	6437.09	4.74	149
	6104.51	6488.62	5.92	154
	6057.7	6250.78	3.09	91
	6512.62	6638.18	1.89	175
	6076.86	6453.73	5.84	57
	6316.49	6757.32	6.52	74
	5942.67	6367.95	6.68	87
	6861.61	7131.83	3.79	135
200	7332.27	7652.29	4.18	160
	6951.85	7328.86	5.14	180
	7570.3	7975.45	5.08	122
	6931.8	7256.95	4.48	280
	6988.3	7330.26	4.67	137
	7116.71	7446.13	4.42	180
	7157.41	7576.16	5.53	119
	7376.74	7987.2	7.64	88
	6510.11	6758.54	3.68	141
	7005.24	7336.87	4.52	210

Table 13 Lagrangian Heuristic Results for Dataset 2 ($N = 400 - 500$)

N	Lagrangian Heuristics			
	LB	UB	% Gap	Time
400	15360.9	16058.6	4.35	285
	15647.8	16065.9	2.60	155
	15895.3	17129	7.20	209
	15912.5	16512.7	3.63	189
	14915.9	15735.4	5.21	542
	16421.4	17035.6	3.61	155
	15339.8	16366.3	6.27	218
	15455.9	16315.7	5.27	164
	15474.1	16403.4	5.67	243
	14660.6	15104.7	2.94	187
450	18349.4	18934.4	3.09	347
	18913.2	19701.6	4.00	154
	16354.7	16905.2	3.26	230
	18111.1	18664.9	2.97	227
	17847.2	18317.5	2.57	157
	18178.1	18970.5	4.18	492
	16953.1	17418	2.67	270
	16380.4	16974.1	3.50	168
	16855.6	17753.1	5.06	279
	17784.2	18631.1	4.55	232
500	18503.2	19078.2	3.01	401
	18296.1	19651	6.89	369
	19119.1	20009.1	4.45	186
	18788.9	19365.8	2.98	232
	18605.7	19235	3.27	407
	18821.2	19979.1	5.80	392
	20614.8	21331.4	3.36	1628
	18622.6	19848.1	6.17	288
	19315.8	19921.6	3.04	181
	20356	20917	2.68	184

Table 14 Lagrangian Heuristic Results for Dataset 3 ($N = 800 - 1000$)

N	Lagrangian Heuristics			
	LB	UB	% Gap	Time
800	36245.2	37136.5	2.40	240
	29758.7	30597.9	2.74	1335
	32793.6	33962	3.44	1515
	33753.3	34978.8	3.50	1191
	30682.4	32136.9	4.53	393
	32523.1	33492.3	2.89	397
	29872.1	31308.9	4.59	2591
	33769.4	34683.1	2.63	842
	30644.8	32018.6	4.29	1937
	33711.6	34908	3.43	1159
900	34657.6	37703.6	8.08	386
	34153	35411.1	3.55	500
	37839.1	39185.5	3.44	4190
	39256.4	40665.5	3.47	4441
	33207.6	34803.3	4.58	3830
	35669.9	36660.2	2.70	681
	33849.6	35450.4	4.52	2260
	34946	36143.2	3.31	5066
	33407	34335.2	2.70	678
	34723.2	35815.1	3.05	4646
1000	40232.6	40923.5	1.69	7389
	43433.5	44184.4	1.70	4161
	40580.9	41915	3.18	1026
	40519.6	42002.5	3.53	1056
	41280.5	42569.1	3.03	3457
	41821.7	43711.8	4.32	468
	38405.2	39218.3	2.07	200
	39236.4	40433.6	2.96	5370
	40058.7	41523.6	3.53	6032
	38718.5	40012	3.23	6036

Next, we solved larger problems with the Lagrangian heuristic and we report their performance in Tables 12, 13, 14. The Table 12 presents the Lagrangian heuristic performance on dataset 2 having $N = 125$ to $N = 200$ which has average runtime between 110 and 160 seconds with an average termination percentage gap ranging between 3 to 4 percent. Similarly, as shown in Table 13 and 14, the average gap is still within 3 to 4 percent.

It is worth noting that as opposed to CPLEX, the average gap for the Lagrangian heuristics is consistent even for large problems and the increase in runtime is within acceptable limits. Note that TNDP needs to be solved less frequently than ONDP, therefore, slightly higher runtimes for larger problems are acceptable.

IV.5. Summary and Conclusions

In this chapter, we considered a tactical network design problem with explicit consideration of loads consolidation. We consider a network with multi-commodity flows where each commodity is defined by its unique pair of origin and destination nodes and a known required flow amount. The system is operated in such a way that the commodities are collected and consolidated into truckloads at consolidation centers, a linehaul transfer takes place for the consolidated loads, which are deconsolidated at deconsolidation centers and from there, the commodities are shipped to their final destinations. These type of applications are often found in intermodal and LTL transportation businesses.

We provided an integer programming formulation for the problem. We make an observation that the model resembles the single sourcing capacitated location problem with staircase capacity, which is an NP-hard problem. Since modern state-of-the-art commercial codes such as CPLEX fail to prescribe solution for even small problem instances, one has to look for heuristic methods for efficient solutions for larger problems instances. One of the issues with heuristics and metaheuristic methods is that they provide solutions but do not guarantee bounds on the quality of the solutions. In this chapter, we investigate methods of finding upper and lower bounds on optimal solutions to tactical network design problems.

We modified the compound neighborhood search procedures to adapt to the tactical problem and developed a simulated annealing based heuristic algorithm to provide a feasible solution, which is an upper bound. As opposed to the one-exchange and one-move based neighborhoods in TNDP, we develop two-exchange and two-move based simple neighborhoods that provide good quality solutions in TNDP, despite the fact that there is a smaller number of simple content-change neighborhoods (two as opposed to five in ONDP). With simulated annealing-based metaheuristic, we implement a sophisticated bias tree search strategy which proves to be very efficient in guiding the search in a direction with potentially good solutions.

Although there are several methods for finding lower bounds (e.g. LP based lower bounds), we found that the Lagrangian relaxation based methods provides bound of good quality. We consider two Lagrangian relaxations and showed that the relaxation obtained by dualizing demand satisfaction constraints provides tighter lower bounds and facilitates development of efficient solution algorithms. We also suggest a variable relaxation idea to speed up the computation of Lagrangian relaxation based lower bounds without compromising the solution quality.

Finally, we developed a Lagrangian heuristic framework that utilize an upper bound found by a simulated annealing based metaheuristic to update the Lagrange multipliers using sub-gradient optimization. The Lagrangian framework also utilizes the lower bound solution as input to the simulated annealing method to find even better upper bounds.

The computational study confirms the superiority of the Lagrangian relaxation based lower bound over lower bounds generated based on relaxation of the LP and the LP with tightening constraints. We compared our Lagrangian heuristics lower and upper bounds against CPLEX on benchmark problems. The Lagrangian heuristics provided good quality solution with significantly smaller runtime. For larger problems also the Lagrangian heuristic consistently provided good quality solutions in short runtime.

CHAPTER V

STRATEGIC NETWORK DESIGN PROBLEM

The strategic network design problem (SNDP) is concerned with the resources that provide the foundation for tactical and operational level planning problems investigated in the previous two chapters. Recall that ONDP addresses operational decision issues such as commodity-truck assignments and truck-link assignments assuming that the human resources, equipment and truck capacities are available at the locations prescribed by its solution. This assumption is made possible because the capacity issues are taken care of by a tactical problem that we call TNDP. The TNDP addresses the capacity planning decisions that ensure the availability of resources at the operational level at regular prices i.e. without the need for expedited capacity building at higher prices. However, not all resources can be planned for tactical time horizons. Some examples of such exceptions are buildings or equipment requiring intensive capital investment. We consider these strategic decisions in this chapter. More specifically, in addition to the design issues addressed in TNDP, SNDP also considers the location and capacities of the consolidation/deconsolidation centers and theie associated costs. These centers support operations such as sorting, loading and unloading that require human resources, equipment, buildings and other infrastructure. We assume a base capacity representative of truck handling capacity (loading, unloading, and other auxiliary operations) and let the solution decide the number of base unit capacities to be installed at the centers. The costs associated with the centers varies with location, e.g., buildings and human resources may cost more in California than in Oklahoma. The operations at the consolidation and deconsolidation center nodes may include moving, loading and unloading the trucks.

The formal description of SNDP is as follows. We are given a network with multi-commodity flows as defined in ONDP and TNDP. The system is operated in such a way that the commodities are collected at a consolidation center and sent via consolidated shipments over a linehaul transfer link to a deconsolidation center, and from there the commodities are shipped to their final destinations. The decisions considered in SNDP include

1. the locations and capacities of the consolidation and deconsolidation centers;
2. the linehaul transfer links and their capacities in terms of the number of truckload trips between the consolidation and deconsolidation centers
3. the assignment of commodities to consolidation and deconsolidation centers and, in turn, to transfer links.

The costs in the system include collection costs, linehaul transfer costs, distribution costs, and costs for locating consolidation and deconsolidation centers. We assume that the capacity installments on the linehaul transfer links and at the consolidation and deconsolidation centers are set in fixed increments. On transfer links, capacity can be installed with increments of truckload capacity and at centers, with increments of some base capacity, both with their associated increment costs. Next, we present a mixed integer programming formulation for SNDP.

V.1. The Model

We present a mathematical formulation for the problem using the notation given below:

Notation:

w_i the amount of flow for commodity p_i

σ base capacity (handling number of truckloads trips) at consolidation and deconsolidation centers

U capacity per TL for long-haul transfers

V_j^J cost of installing base capacity σ at consolidation centers j

V_k^K cost of installing base capacity σ at deconsolidation centers k

β full TL transportation cost per mile between $\mathcal{J}$ and $\mathcal{K}$

α^f LTL transportation cost per unit per mile between $\mathcal{F}$ and $\mathcal{J}$

α^t LTL transportation cost per unit per mile between $\mathcal{K}$ and $\mathcal{T}$

d_{ij}^f distance between f_{p_i} and consolidation center j

d_{ki}^t distance between deconsolidation center k and t_{p_i}

d_{jk} distance between centers j and k

T_{jk} $\beta\, d_{jk}$

W_{ijk} $w_i(\alpha^f\, d_{ij}^f + \alpha^t, d_{ki}^t)$

Decision Variables:

z_{ijk} fraction of commodity i's demand assigned to link (j, k)

y_{jk} capacity (number of truckload trips) installed on the link (j, k)

v_j^J number of σ units of capacity installed at consolidation center j.

v_k^K number of σ units of capacity installed at deconsolidation center k.

Objective & Constraints:

$$\text{Min} \quad \sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} W_{ijk} z_{ijk} + \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} T_{jk}\, y_{jk} + \sum_{j\in\mathcal{J}} V_j^J v_J^J + \sum_{k\in\mathcal{K}} V_k^K v_k^K. \tag{5.1}$$

subject to

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} z_{ijk} = 1 \qquad \forall i. \tag{5.2}$$

$$\sum_{i\in\mathcal{P}} w_i z_{ijk} \ \leq U y_{jk} \qquad \forall j,k. \tag{5.3}$$

$$\sum_{k\in\mathcal{K}} y_{jk} \ \leq \sigma v_j^J \qquad \forall j. \tag{5.4}$$

$$\sum_{j\in\mathcal{J}} y_{jk} \ \leq \sigma v_k^K \qquad \forall k. \tag{5.5}$$

$$z_{ijk} \geq 0, y_{jk}, v_j^J, v_k^K \in \mathcal{Z}^+ \qquad \forall i,j,k. \tag{5.6}$$

The first term in the objective function (5.1) shown above is the sum of the collection and distribution costs for all the commodities, The last three terms represent the sum of costs of capacity installations on transfer links, on consolidation centers and de-consolidation centers, respectively. The first set of constraints (5.2) makes sure that each commodity's demand is satisfied. The next three sets of constraints (5.3), (5.4), and (5.5) impose a capacity restriction on every transfer link, consolidation center and de-consolidation center, respectively. Finally, constraint (5.6) implies nonnegativity restrictions on $\mathbf{z}$ and nonnegativity and integrality restrictions on $\mathbf{y}, \mathbf{v^J}, \mathbf{v^K}$.

Some remarks regarding the model are in order. The model for SNDP shown here builds on the model for TNDP by simply incorporating location and capacity decision variables relating to the consolidation and deconsolidation centers. Notice that the assignment variable $\mathbf{z}$ is defined as a fraction of commodity i's demand assigned to the link (j, k) as opposed to the case of ONDP and TNDP where single allocations are considered. The multiple allocation of a commodity is motivated by practical considerations. Single sourcing makes practical sense for operational reasons, but SNDP deals with time horizons longer than ONDP and TNDP, and therefore, although at the operational level, it may be a business requirement to assign commodity to exactly one source, that source may change over a period of time. For example, consider a consumer goods store which, during the winter season, sources woolen clothes from China and, therefore, allocates stores in California to a distribution center at Los Angeles. In summer, however, it sources supplies from South America and wishes to assign the same stores to a distribution center in Phoenix during the summer season. These assignments are single source within each season, but in a time horizon that includes both seasons, the store allocation is multiple. Clearly, single assignment is not the right choice in the case of strategic level network design. Moreover, having multiple allocations is a robust design from the reliability perspective also. Although we do not model risk and uncertainty here, needless to say, it could be an interesting extension of the problem.

Notice that although the assignment decision variable is defined as a fraction, we do not need to specify the upper bound as 1 because it is implied by constraint (5.2). The problem has a structure that the constraints (5.4) and (5.5) involve involve the integer variables whereas the constraint (5.2) involved only fractional variable. The constraint (5.3) is the linking constraint between the assignment variable z and the center capacity variables v^j and v^k. For fixed values of the integer variables, this problem is nothing but a transportation problem. This type of structure can be exploited in developing solution algorithms as we will see shortly. Also, we can replace constraint (5.2) by $\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} z_{ijk} \geq 1, \forall i$ which is a relaxation because of the inequality, however it is tight, because it provides exactly the same solution.

To the best of our knowledge SNDP has not been solved in the literature. SNDP is an interesting problem that incorporates features from the network design and facility locations area. Not only does it model the node capacity as a constraint, but also as a decision variable. The problems most closely related to SNDP are multisourcing staircase capacitated facility location problems and point to point delivery network design problems, the literature related to which was discussed in the Chapter II. In the following section, we develop a Benders decomposition-based solution approach for SNDP.

V.2. Benders Decomposition Based Solution Approach

Benders decomposition has been successfully applied to many combinatorial optimization problems. In general terms, the Benders decomposition technique involves decomposing the main problem into a master and a subproblem, and then solving them iteratively by utilizing the solution of the one in the other. The first problem is called the Benders master problem, which involves solving a problem obtained by removing some of the constraints and obtaining the solution of a subset of variables. The values of the variables fixed in the master problem, are then substituted in the second problem called the Benders subproblem. The solution to this Benders subproblem is used to fix the values of the remaining variables and to generate a cut for the Benders master problem. This process is repeated until a termination condition, usually a time limit or maximum maximum number of iterations, is met. Since the decomposed problems need to be solved at each iteration, it is desirable to have subproblems that can be solved easily. The two areas that closely relate to SNDP are network design problems and capacitated facility location problems. The Benders decomposition approaches developed for these problem areas exploit the structure of the problem, i.e. the subproblems obtained in these problems are often shortest path, transportation type problems which are easy to solve. Geoffrion and Graves (1996) developed a Benders decomposition-based solution method for multicommodity distribution system design and implemented it for a major food firm. In this case, when the primal variables are fixed, the subprobelm is the classical transportation problem that can be further separated by commodities. The fixed charge network design problem has a structure suitable for Benders decomposition, because the arcs selection variable (link open/close) is solved in the master problem and the actual flow can be solved in the subproblem. Refer to a recent review by Costa (2005) on Benders decomposition applied to fixed-charge network design problems. Similarly, Capacitated facility location problems also naturally decompose in such a way that the facility location decision variables are solved in the master problem whereas customer assignment is solved in the Benders subproblem. Roy (1986) develops a cross decomposition-based approach for a capacitated facility location problem which combines Lagrangian and Benders decomposition in a single framework to iteratively fix the location variables and solve the resulting transportation problem to produce a new lagrangian multiplier. For fixed lagrangian multipliers, they solve the Lagrangian relaxation to produce new values for the location variables. Magnanti and Wong (1981) and Roy (1986) shown some methods to tailor the Benders method to perform more efficiently by generating stronger cuts.

Our SNDP formulation employs integer variables $\mathbf{y}, \mathbf{v^J}, \mathbf{v^K}$ to model capacity related decisions and the continuous variable $\mathbf{z}$ for allocation decisions. We observe that for known capacities (for fixed $\mathbf{y}, \mathbf{v^J}, \mathbf{v^K}$) the allocation problem is a linear program which can be solve efficiently. Similarly, the problem excluding allocation decisions is an integer program involving much smaller number of variables and constraints (only

M^2 + 2M variables and 2M constraints) which is easier to solve. We develop a Benders decomposition-based algorithm to solve SNDP that exploits this structure. We call the problem of finding best allocations for fixed $\mathbf{y}, \mathbf{v^J}, \mathbf{v^K}$ as Benders subproblem (BSP). Clearly, the sum of optimal solution to the BSP and the fixed cost associated with $\mathbf{y}, \mathbf{v^J}, \mathbf{v^K}$ provides a valid upper bound to the optimal solution of the original problem. The Benders decomposition technique reformulates the original problem by introducing a continuous variable and a cut utilizing the information (dual variables) form BSP. This reformulation of the original problem is called Benders Master Problem (BMP). The optimal solution of the BMP is a valid lower bound on the optimal solution of the original problem. The solution of the BMP is used again in solving the BSP. In practice, the convergence of the Benders decomposition-based approaches depend strongly on the quality of the Benders cuts that are added to the BMP. Another advantage of the Benders decomposition method over other metaheuristic and heuristics methods is that it provides both lower and upper bounds whereas heuristics provide only a feasible solution, but not a bound for the solution quality.

Above, we only presented a brief overview of Benders decomposition technique, for a comprehensive discussion we refer the reader to the text book by Daskin (1995) and the review by Costa (2005).

In the section that follows, we describe SNDP specific Benders subproblem, its dual, the Benders cut, and the Benders master problem followed by a method for generating stronger Benders cuts and finally we provide the overall algorithmic framework for the Benders decomposition algorithm for SNDP.

V.2.1. Benders Subproblem

Let $\bar{\mathbf{y}}, \bar{\mathbf{v}}^{\mathbf{J}}$ and $\bar{\mathbf{v}}^{\mathbf{K}}$ represent the given capacity installations on links and centers that guarantee feasible allocations. Then, given the capacities that provide a feasible assignment, the Benders subproblem LP is simply a transportation problem given by **BSP-LP** as shown below:

$$\text{Min} \quad \sum_{i\in\mathcal{P}}\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} W_{ijk} z_{ijk} \tag{5.7}$$

subject to

$$\sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} z_{ijk} = 1 \qquad \forall i. \tag{5.8}$$

$$\sum_{i\in\mathcal{P}} w_i z_{ijk} \le U\bar{y}_{jk} \qquad \forall j,k. \tag{5.9}$$

$$z_{ijk} \ge 0 \qquad \forall i,j,k. \tag{5.10}$$

The objective function (5.7) represents the total cost of allocation, i.e. the sum of the collection and distribution costs for all commodities. Constraints set (5.8) ensures that every commodity's demand is satisfied, and constraints set (5.9) makes sure that total demand assigned to each link is less than, or equal to, the capacity installed on it. We observe that the subproblem is a transportation problem which can be solved easily.

$$z_{ijk} \le \bar{y}_{jk} \qquad \forall i,j,k. \tag{5.11}$$

The BSP can be tightened by adding constraint set (5.11) which dictates that a commodity can be assigned to a link only if a nonzero capacity is installed on it. These are valid inequalities. Although not required for correct formulation of the problem, they define a tighter polyhedral description of the solution space. The optimal objective function value of **BSP-LP** added to the cost associated with fixed value of $(\mathbf{y}, \mathbf{v^J}, \mathbf{v^K})$ as given by (5.26) provides an upper bound on the original problem.

Defining $\mu_i,\ \delta_{ijk},\ \lambda_{jk}$ as the dual variables for constraints (5.8), (5.11) and (5.9), respectively, the equivalent dual of the **BSP-LP** is given by **BSP-DUAL** as shown below:

$$\text{Max} \quad \sum_{i \in \mathcal{P}} \mu_i + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left(- \sum_{i \in \mathcal{P}} \delta_{ijk} - U\lambda_{jk} \right) \bar{y}_{jk} \tag{5.12}$$

subject to

$$\mu_i - \delta_{ijk} - w_i \lambda_{jk} \le W_{ijk} \qquad \forall i, j, k. \tag{5.13}$$

$$\mu_i\text{-free}, \delta_{ijk}, \lambda_{jk} \ge 0 \qquad \forall i, j, k. \tag{5.14}$$

Let $\mathcal{B}$ denote the set of all extreme points of the **BSP-DUAL** given by (5.12)-(5.14). Also, we represent by $(\mu_i, \delta_{ijk}, \lambda_{jk})^b$ and η^b the dual variables and objective function values, respectively, that are associated with an extreme point $b \in \mathcal{B}$. Let η^* represent the objective value corresponding to the optimal extreme point. Then, we must have $\eta^* \ge \eta^b\ \forall b \in \mathcal{B}$, where

$$\eta^b = \sum_{i \in \mathcal{P}} \mu_i^b + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left(- \sum_{i \in \mathcal{P}} \delta_{ijk}^b - U\lambda_{jk}^b \right) \bar{y}_{jk} \tag{5.15}$$

Clearly, then we can write the dual problem in terms of extreme points as

$$\min_{\eta \ge 0} \quad \eta \tag{5.16}$$

subject to

$$\eta \geq \sum_{i \in \mathcal{P}} \mu_i^b + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left(-\sum_{i \in \mathcal{P}} \delta_{ijk}^b - U\lambda_{jk}^b \right) \bar{y}_{jk} \qquad \forall b \in \mathcal{B} \tag{5.17}$$

V.2.2. Benders Master Problem

Knowing the **BSP-DUAL** problem representation using the extreme points, we can reformulate the original problem, called the Benders Master Problem (**BMP**) as shown below.

$$\text{Min } \eta + g(\mathbf{y}, \mathbf{v^J}, \mathbf{v^K}) \tag{5.18}$$

subject to

$$\eta \geq \sum_{i \in \mathcal{P}} \mu_i^b + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left(-\sum_{i \in \mathcal{P}} \delta_{ijk}^b - U\lambda_{jk}^b \right) y_{jk} \qquad \forall b \in \mathcal{B} \tag{5.19}$$

$$\sum_{k \in \mathcal{K}} y_{jk} \leq \sigma v_j^J \qquad \forall j. \tag{5.20}$$

$$\sum_{j \in \mathcal{J}} y_{jk} \leq \sigma v_k^K \qquad \forall k. \tag{5.21}$$

$$\sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} U\, y_{jk} \geq W \tag{5.22}$$

$$\sum_{j \in \mathcal{J}} U \sigma v_j^J \geq W \tag{5.23}$$

$$\sum_{k \in \mathcal{K}} U \sigma v_k^K \geq W \tag{5.24}$$

$$\eta \geq 0, y_{jk}, v_j^J, v_k^K \in \mathcal{Z}^+ \qquad \forall i, j, k. \tag{5.25}$$

where the term $g(\mathbf{y}, \mathbf{v^J}, \mathbf{v^K})$ represents the fixed cost of capacity installations on linehaul transfer links and fixed cost of locating centers as shown below in Equation (5.26).

$$g(\mathbf{y}, \mathbf{v^J}, \mathbf{v^K}) = \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} T_{jk} y_{jk} + \left(\sum_{j \in \mathcal{J}} V_j^J v_J^J + \sum_{k \in \mathcal{K}} V_k^K v_k^K \right) \tag{5.26}$$

The BMP formulation shown above contains a large number of constraints (as many as the extreme points in the **BSP-DUAL**) that implicitly contains the assignment part of the problem. Notice that the surrogate constraints (5.22), (5.23) and,(5.24) which were not present in the main formulation before, are not essential for correct formulation of the problem. They imply that the total capacity installed on each link, consolidation center and deconsolidation center meets or exceeds the total demand.

The problem with this reformulation is that it involves large numbers of constraints, one for each extreme point, which makes it prohibitively large to solve. Moreover, at optimality, many of the constraints that correspond to extreme points will be satisfied. Therefore, in the Benders decomposition method, one works with only a restricted set of constraints. The master problem that includes constraints from the restricted set of extreme points is called the Restricted Master Problem (RMP). Instead of adding the constraints that correspond to the extreme points *all at once*, this algorithm adds them iteratively. Each dual subproblem is used to generate a cut which is added to the master problem to obtain a new set of dual variables. The surrogate constraints are essential so that the capacities $\bar{\mathbf{y}}, \bar{\mathbf{v}}^{\mathbf{J}}, \bar{\mathbf{v}}^{\mathbf{K}}$, prescribed by solution of the Benders master problem, guarantee a feasible assignment. Next, we summarize the Benders cut and an algorithm to generate stronger Benders cuts.

V.2.3. Strong Benders Cut

Let us denote the optimal dual variables by μ_i^*, δ_{ijk}^* and λ_{jk}^*, then the benders cut is given by

$$\eta \geq \sum_{i \in \mathcal{P}} \mu_i^* + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left(T_{jk} - \sum_{i \in \mathcal{P}} \delta_{ijk}^* - U \lambda_{jk}^* \right) y_{jk} \tag{5.27}$$

At each iteration, we obtain a new dual solution and substitute it in the equation (5.27) and then add it to the RMP and re-solve.

It has been observed that when a Benders subproblem has multiple dual optimal solutions, the optimal dual reported as the optimal solution to the **BSP-DUAL** may not lead to the strongest cut as shown in equation (5.27). Degeneracy is a very well known issue in transportation problems, and in most cases, there exist multiple dual optimal solutions. Magnanti and Wong (1981) and Roy (1986) propose an algorithm for deriving stronger benders cut.

A cut Π^1 is said to dominate the other cut Π^2 if all points that satisfy Π^2 also satisfy Π^1. However there exists at least one point which satisfies Π^2 but not Π^1. In the presence of multiple optimal dual solutions, in order to find the strongest cut, one needs to identify the optimal dual variable corresponding to the strongest cut.

Next we explain an algorithm to strengthen the Benders cut. The algorithm tries to strengthen the cut given by (5.27) which is reproduced below for convenience.

$$\eta \geq \sum_{i \in \mathcal{P}} \mu_i^* + \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \left(T_{jk} - \sum_{i \in \mathcal{P}} \delta_{ijk}^* - U \lambda_{jk}^* \right) y_{jk} \tag{5.28}$$

Notice that in the cut given above, when $\bar{y}_{jk} = 0$, one can modify its coefficient $\left(T_{jk} - \sum_{i \in \mathcal{P}} \delta_{ijk}^* - U \lambda_{jk}^*\right)$ without changing the objective function value, provided feasibility is maintained, i.e. it satisfies constraint (5.13). This can be achieved by solving the following LP.

$$\text{Max } T_{jk} - \sum_{i \in \mathcal{P}} \delta_{ijk} - U\lambda_{jk} \tag{5.29}$$

subject to

$$\mu_i - \delta_{ijk} - w_i\lambda_{jk} \leq W_{ijk} \qquad \forall i. \tag{5.30}$$

$$\delta_{ijk}, \lambda_{jk} \geq 0 \qquad \forall i. \tag{5.31}$$

In this LP, the objective is the coefficient of y in the cut given by (5.27) and the constraints are the same as in **BSP-DUAL** . This LP must be solved for every (j,k) s.t. $y_{jk} = 0$ to obtain the dual multipliers that give the strong cut.

V.2.4. Benders Decomposition Framework

Having defined the Benders subproblem **BSP-LP**, **BSP-DUAL** ,**BMP** and the algorithm to generate a stronger cut, below we describe the Benders decomposition-based algorithm in Display 9. The algorithm starts with initializing the integer variables by solving the master problem (5.18)-(5.25) without any cut. We represent this initial solution by $(\mathbf{y}, \mathbf{v^J}, \mathbf{v^K})^0$. The basic framework consists of iteratively solving the **BSP-LP** or **BSP-DUAL** using the primal solution $(\mathbf{y}, \mathbf{v^J}, \mathbf{v^K})^{itr}$ at each iteration itr, and updating the incumbent upper bound if improvement is observed, obtaining the dual variables $(\mu^*, \delta^*, \lambda^*)^{itr}$ and generating a strong cut using the method as described in line 9 of the Display. After generating the Benders cut, we solve the **BMP** to obtain new primal variables and continue the process until termination, as specified in line 2 is met. This basic framework may be tuned for the specific dataset for performance. We observe that the optimal solution to **BMP**, which is a lower bound on the original problem, often takes longer as the number of cuts is increased due to growth in the problem size. This affects the performance of the algorithm in an adverse manner. We allow CPLEX to solve **BMP** only to the optimality gap of 0.01 percent or 25 seconds, whichever occurs first. Upon termination, we record the lower bound to the **BMP** and pass it on as the LB^{itr}. This helps to keep the algorithm runtime in check and at the same time provides reasonable quality solutions.

Upon termination, the algorithm returns the objective value and the feasible solution.

V.3. Computational Results

In order to test the efficiency of our solution method, we tested our method on randomly generated problems as well as data having characteristic from a large parcel company is US.

Display 9 Benders Decomposition Based Algorithm

initialize `MAXITER`=100, `itr`=0, gap=0

LB$=-\infty$, UB$=\infty, \epsilon = 0.1$

$\mu^* = 0, \delta^* = 0, \lambda^* = 0$

$\Psi = \emptyset$

$(\mathbf{y}, \mathbf{v^J}, \mathbf{v^K})^0 = \arg\min\{\sum_{j\in\mathcal{J}} \sum_{k\in\mathcal{K}} T_{jk} y_{jk} + \sum_{j\in\mathcal{J}} V_j^J v_j^J$
$+ \sum_{k\in\mathcal{K}} V_k^K v_k^K \ s.t. \ (5.20) - (5.24)\}$

while (`itr` $\leq$ `MAXITER` **AND** `gap` > ϵ) **do**

  itr = itr + 1

  Obtain UB^{itr} and $(\mu^*, \delta^*, \lambda^*)^{itr}$ from (5.12)-(5.14) and (5.26)

  if UB^{itr} < UB **then**

    UB = UB^{itr}

  end if

  Generate strong cut (5.27) using (5.29)-(5.31)

  Add the cut (5.27) to set of cuts Ψ

  Obtain LB^{itr} and $(\mathbf{y}, \mathbf{v^J}, \mathbf{v^K})^{itr}$ by solving (5.18)-(5.25)

  if LB^{itr} > LB **then**

    LB = LB^{itr}

  end if

  gap $= 100 \times \frac{UB-LB}{UB}$

end while

RETURN UB and $(\mathbf{z}, \mathbf{y}, \mathbf{v^J}, \mathbf{v^K})$

V.3.1. Experimental Setup

The process of generating random test instances is given on page 15. In addition, we generate test instances using the data from a real life application, a large parcel company in the US. Typically, large small-package carriers and LTL shipment operators strive to provide services to cover a large area, such as anywhere in the US. Although the demand (outgoing and incoming) for service may not be equally large in all locations, it may be desirable for the company to cover an entire area. As we mentioned earlier, the flow between the various origin and destination pairs may show a lot of variation. For example, there may be flow of over a million packages from New York City to Houston while the flow from College Station to Boston may be only in the hundreds. To capture this feature , we have used a large parcel company's data to determine the parameters. We generate the demands (w) as:

$$w = \begin{cases} 70\% & \text{of the demand is uniformly distributed between 0.10 and 0.20 units} \\ 20\% & \text{of the demand is uniformly distributed between 0.30 and 0.60 units} \\ 10\% & \text{of the demand is uniformly distributed between 0.10 and 0.90 units} \end{cases}$$

Apart from the demand, we use the truck capacity U to be 8 units, the base capacity for centers σ to be 2 truckload trips, values of α^f and α^t as 1 per unit per mile, the TL shipment rate β as calculated on page 15, should be 6 per mile. We further reduce β to 5 per mile to account for discount pricing from the suppliers for long term quantity commitment and negotiations. The complete list of problems and settings to generate them is shown in Table 15. The Benchmark Problem are solved using CPLEX as well as our algorithm. Dataset 1, 2 and 3 involve problems of increasing sizes. The dataset 4 and 5 have the problems generated-based on real application data.

Table 15 SNDP Experimental Problem Sets

Datasets	E	N_P	N	M	U
Benchmark Problems	25	25	$100, 200, 300, 400$	10	8
Dataset 1	40	40	$500, 600, \ldots, 1200$	6	8
Dateset 2	50	40	$1300, 1400, 2000$	6	8
Dataset 3	100	100	$3000, 4000$	6	8
Dataset 4	100	100	$3000, 4000$	10	8
Dataset 5	100	100	1000	25	8

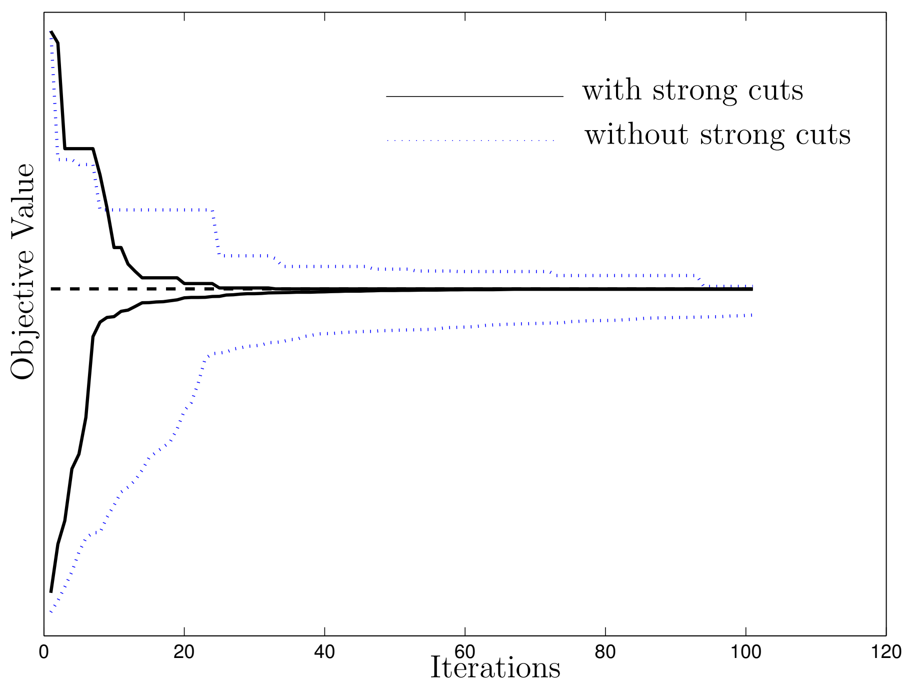
V.3.2. Computational Results

In order to evaluate the impact of adding the strong cuts, we solved small problems with the algorithm as shown in Display 9, with and without strong cuts. We compare the two cases in Figure 15. In Figure 15, we show iterations on the x-axis and the value of the upper and lower bounds on the y-axis. The blue dotted line shows the convergence trend for the algorithm without strong cuts, and the black solid line shows the convergence trends for algorithm with strong. Clearly, the stronger cuts are very effective in speeding up convergence and they affect both the upper bound as well as the lower bound. In all of the following computational experiments, we always use strong cuts.

We were unable to obtain solutions to even small problem instances using the the formulation (5.1) - (5.6). Therefore we decided to include the valid inequality (5.11) and surrogate constraints (5.22)-(5.24). In all of our testing, we have used formulation including these constraints to obtain solutions from CPLEX. We observed that although adding these valid inequalities and surrogate constraints enables us to solve small problem instances, the moderately larger problems still remain intractable to solution by CPLEX due to memory issues.

In Table 16, we benchmark the upper and lower bounds obtained from the Ben-

Figure 15 Banders Decomposition Based Algorithms: With and Without Strong Cuts



ders decomposition-based approach. We used CPLEX to obtain the optimal solution with default options. The lower and upper bounds obtained from the Benders decomposition are closed to the optimal solution. We notice that the upper bound is almost equal to the optimal solution and the lower bound is less than 0.3 percent away from the optimal solution. In addition, the solution time is reasonably short. On an average, for benchmark problems, the Benders termination gap was 0.18 percent with a runtime of 88.95 seconds compared to a termination gap of 0.01 percent in 40.7 seconds for CPLEX. The termination criterion for the Benders algorithm as shown in Display 9 was used, i.e. **MAXITER** $= 100$ and $\epsilon = 0.1$.

Table 16 Benders Bounds for Benchmark Dataset 0

N	% Gap		Time	
	LB	UB	CPLEX	Benders
100	0.09	0.00	6.50	11.10
120	0.09	0.00	8.20	23.50
140	0.10	0.00	8.10	25.80
160	0.12	0.00	12.80	50.20
180	0.16	0.02	21.90	75.00
200	0.22	0.02	31.60	122.90
220	0.16	0.01	19.10	101.00
240	0.19	0.02	54.90	83.70
260	0.20	0.01	58.30	94.00
280	0.18	0.02	45.80	119.90
300	0.19	0.03	35.80	91.50
320	0.21	0.03	93.50	127.40
340	0.22	0.03	149.00	157.00
360	0.27	0.06	82.80	126.10
380	0.16	0.06	7.50	97.00
400	0.25	0.04	30.60	112.80

Table 17 and 18 summarize the computational results for problems from size N = 500 to 2000. We use CPLEX to solve the BMP and BSP and the LP for generating strong cuts. Notice that the BMP is an integer program and with every iteration of the algorithm, it increases in size due to the addition of cuts. We observed that solving the BMP to optimality, takes a lot of computational time, with only marginal gain in the quality of the bound. The method also suffered from the tailing off effect. In other words, CPLEX is able to find solution of reasonable quality (upto 0.1 percent) quickly but then suffers from tail-off effect. Therefore, we limited the time spend on BMP to 25 seconds, and we recorded the best lower bound at termination as our new lower bound obtained form the BMP. We observed significant savings in computation time by this way and obtained reasonably good bounds as seen in Tables 17 and 18.

We observe that Benders algorithm is able to obtain good quality solution, most of them within a 2% gap and a computational time within 4 to 8 minutes for all problems with N up to 2000. In the results presented for N varying from 500 to 2000, the average termination gap is 0.86 percent, with maximum, minimum and standard deviations of 2.45, 0.09 and 0.49 percent, respectively. Similarly, the average runtime is 283.66 seconds, with a maximum of 479 seconds , a minimum of 112 seconds, with standard deviations of 91.71 seconds. Therefore, we conclude that the Benders algorithm developed here provides consistently good quality solutions within small amounts of time.

Table 17 Benders Bounds for Larger Problem Instances N=500-1200

N	% Gap	Time	N	% Gap	Time
	0.17	119		0.35	195
	0.60	271		0.75	209
	0.39	140		0.96	217
	0.67	460		0.66	217
500	0.44	176	900	0.83	197
	0.34	144		0.46	201
	0.40	130		1.36	479
	0.12	112		0.92	213
	0.73	402		1.00	224
	0.20	125		0.73	210
	0.10	132		0.74	221
	0.42	128		0.09	190
	0.10	118		1.44	379
	0.17	138		0.70	239
600	0.43	129	1000	0.81	235
	0.24	129		0.64	204
	0.65	280		0.39	208
	0.22	135		0.55	224
	0.39	137		0.82	218
	0.36	148		0.49	224
	0.45	172		0.59	205
	0.66	193		1.20	250
	0.43	169		0.31	242
	0.40	164		0.42	239
700	0.14	158	1100	0.25	242
	0.36	160		0.45	237
	0.12	148		1.31	283
	0.78	231		0.99	237
	0.39	156		0.16	251
	0.28	156		0.73	245
	0.90	237		1.11	269
	0.43	173		1.47	276
	0.49	176		0.83	249
	0.45	181		1.10	258
800	0.20	181	1200	1.63	294
	1.04	357		1.23	255
	0.59	219		1.41	263
	0.58	226		0.59	261
	0.65	162		1.28	249
	0.09	171		0.67	222

Table 18 Benders Bounds for Larger Problem Instances N=1300-2000

N	% Gap	Time	N	% Gap	Time
	1.25	284		1.12	370
	1.01	285		1.61	388
	1.45	299		0.81	377
	0.64	288		1.95	379
1300	1.03	281	1700	1.43	370
	1.00	290		0.62	372
	0.91	286		1.09	375
	0.92	289		1.59	366
	0.77	287		1.57	373
	1.14	290		1.59	371
	1.07	310		0.60	375
	0.51	306		0.66	391
	1.51	312		0.40	396
	0.92	297		0.63	396
1400	1.12	307	1800	0.65	395
	0.91	308		1.23	382
	1.54	320		0.50	386
	1.04	308		0.57	394
	1.04	307		1.85	391
	1.21	284		1.71	366
	1.01	320		0.25	391
	1.17	324		0.59	391
	0.10	341		2.28	381
	0.59	325		0.76	376
1500	1.11	327	1900	1.07	389
	0.77	324		1.57	386
	0.83	322		1.37	376
	0.95	327		2.45	360
	0.39	327		1.05	398
	0.92	331		1.59	382
	1.23	342		0.72	409
	1.46	349		1.19	400
	2.02	364		1.31	395
	0.91	352		1.56	408
1600	2.22	389	2000	0.81	423
	1.08	340		0.53	425
	0.84	355		0.71	425
	1.24	347		2.01	427
	1.17	353		1.19	428
	1.15	301		1.93	430

Table 19 Benders Bounds for Larger Problem Instances N=3000 and 4000

N	LB	UB	% Gap	Time (Seconds)
	253220	258497	2.04	646
	255006	257580	1.00	559
	250036	252541	0.99	639
	276891	279590	0.97	605
3000	275970	284406	2.97	962
	248971	252611	1.44	1192
	271902	274619	0.99	472
	266465	275224	3.18	545
	267080	277202	3.65	556
	256964	262000	1.92	546
	317380	323667	1.94	1125
	322776	336906	4.19	1526
	334981	339693	1.39	919
	347436	356398	2.51	727
4000	323077	330742	2.32	751
	354362	363008	2.38	1037
	312763	316899	1.31	1045
	338618	341879	0.95	664
	320963	324191	1.00	1224
	347356	355081	2.18	1350

Table 19 presents the largest problems that we attempted to solve. As before, we limited the Benders algorithm to a maximum of 100 iterations. The BMP is solved to optimality within 0.1% with a maximum runtime of 25 seconds. Again, our algorithm has been able to find good quality solutions with an average gap of 1.97 percent and an average runtime of 854.5 seconds. 9 out of 10 instances with $n = 3000$ take less than 1000 seconds, and similarly, the run times are well within reasonable limits in the cases of $N = 4000$.

In Table 20, we present the solutions to the problems instances generated based on real data from a parcel company. Instead of having two separate squares for origin and destination nodes, in this problem set, we assume a single square. Again, the Benders solves large problems of $N = 2000$ and 3000 with no difficulty. The gaps at termination are again within 3 percent. The runtime for these problems is, however, larger than those in the previous datasets but still within reasonable limits.

Finally, the third set of results in Table 20 involves M as 25, which increases the problem size significantly. Just to get an estimate of the increase in size from M=10 to M=25, for N=1000, the number of variables in **BSP-LP** increases from 100,000 to 625,000. Similarly, the size of the BMP also increases significantly. We observe that the gaps at termination are below 2 percent for 8 out of 10 instances, and the remaining two are at 2.07 percent and 2.06 percent. Computational runtime are higher now, but since SNDP is a problem which does not need to be solved on a regular basis, the increased runtime is acceptable.

Table 20 Benders Bounds for Data Similar to Parcel Company

N	LB	UB	% Gap	Time (Seconds)
$M = 10$				
	56911.4	59337.4	4.09	2336
	62017.4	63140.1	1.78	821
	63240.1	64390.7	1.79	990
	61129.2	61811.2	1.10	618
3000	61001.5	62182.1	1.90	974
	59840.4	60610	1.27	728
	60821.2	62151.3	2.14	1159
	59375.1	61133.6	2.88	2112
	66061.9	67515.3	2.15	747
	59802.1	60974.2	1.92	1301
	40249.8	40964.3	1.74	2163
	37848.9	38389.8	1.41	1143
	44711.4	45182.7	1.04	736
	36749.3	37298.5	1.47	2211
4000	41502.2	41990.4	1.16	931
	41571.3	42118.6	1.30	687
	39020.6	39380.4	0.91	1424
	40550.9	41198.3	1.57	1587
	42024.4	42688.9	1.56	1695
	43852.6	44297.3	1.00	434
$M = 25$				
	40249.8	40964.3	1.74	2163
	19508	19921.7	2.08	5025
	18883.9	19081.5	1.04	4759
	18732.6	19046.5	1.65	5936
1000	19015	19257.5	1.26	8394
	17560.1	17930	2.06	8657
	19162.9	19375.4	1.10	7062
	18034	18224	1.04	7297
	18148.4	18447.5	1.62	7590
	18931.5	19183.2	1.31	7276
	18089.4	18344.6	1.39	6162

V.4. Summary and Conclusions

In this chapter we study the strategic network design problem (SNDP), which considers those network design issues considered in the tactical network design problem, and further extends to include location and capacity decisions related to consolidation and deconsolidation centers. The time horizon in the context of SNDP is much longer than the previous problems considered in this dissertation. The strategic network design issues addressed by SNDP provide a foundation for the tactical and operations level business operations. SNDP mainly addresses strategic capacity installation on linehaul transfer links in terms of truckload trips handling capacity of the consolidation and deconsolidation centers (which implies investment in equipment, building and human resources, etc.). Also, SNDP considers allocation of commodities to multiple consolidation and deconsolidation centers, thereby to multiple linehaul links. We assume that the capacity installments on linehaul transfer links and at the consolidation and deconsolidation centers are set in fixed increments. On transfer links, capacity can be installed with increments of truckload capacity and at centers, with increments of some base capacity, both with their associated increment costs.

We presented a mixed integer programming formulation for SNDP using a nonnegative allocation variables (fractions) to model the assignment of commodities to linehaul links and integer variables to model the capacity of transfer links as well as consolidation and deconsolidation centers. Our problem formulation has a structure in which, for fixed capacities, the subproblem reduces to transportation problem, integer program excluding the real variables (the integer program involving only the integer variables) is a relatively small. Such a structure motivates the Benders decomposition method as a solution methodology. We present a study of Benders decomposition developed for applications with similar structures. Problems with similar structure such as capacitated facility location and fixed charge network design problems, have been solved by Benders decomposition methods in the literature.

In the Benders decomposition for our problem, fixing the capacities results in the transportation problem as our subproblem which is a linear program and easy to solve. We find that because of the abundant degeneracy in the transportation problem, we can find the optimal dual variables that generate strong Benders cuts by solving a simple LP. In our empirical testing, we found that the generation of strong cuts helps the Benders decomposition algorithm converge much faster. Although finding optimal dual variables requires additional computation time but it reduces the number of iterations required for convergence and also facilitates better primal solutions by BMP, which, therefore, results in a reduction in overall computation time.

In solving the Benders master problem, we observed that when smaller numbers of cuts are added, CPLEX is able to solve the BMP quickly. However, for larger problems, CPLEX starts to take longer and suffers from tailing off effect. To overcome this problem, we allowed it to run only for limited time and recorded the lower bound at termination as the incumbent lower bound. This strategy seemed to improve runtime without affecting solution quality.

We tested our solution methodology on two sets of data, the first generated randomly using parameters similar to the ONDP and TNDP and the second generated using data from a large parcel company. We found the Benders-based solution method to perform consistently well for small as well as comparatively larger problems. All solutions are within a 2 to 4 percent optimality gap and the run times are also within reasonable limits.

A few remarks on the potential impact of this study are in order. Since tactical and operational level networks rely on a basic infrastructure that is presumed to be available, the solution of SNDP can help a company make decisions possibly at the strategic levels, that backed by analytical models rather than simply soft methods/or guess work. Managers and leaders making strategic plans can use SNDP solutions to negotiate favorable contracts with third party suppliers by offering attractive load commitments and long term partnerships as well as providing better quality service to their customers. Also, better strategic planning results in a smoother tactical and operational level performance by the company and also minimizes the need for the ad hoc and hurried arrangements. Further, in terms of algorithm development, we developed a Benders decomposition-based method for solving large problems. This approach may be extended to solve other service network design problems with similar structure.

CHAPTER VI

CONCLUSIONS AND FUTURE DIRECTIONS

Intermodal transportation and LTL transportation has grown rapidly in the recent past, and this growth is expected to continue due to globalization and increasing economic activity. Growth and change in business has put pressure on industry to be more efficient and productive. Intermodal transportation models are comparatively new (Crainic and Kim, 2005), and the traditional hub-and-spoke network models may not be suitable for addressing the needs of dedicated models for network design problems in the context of intermodal and LTL transportation problems. With this motivation, this dissertation addresses the need for dedicated models and solution methods for intermodal and LTL transportation network design problems. The potential impact of this dissertation can be summarized as follows.

1. The models developed in this dissertation will provide further insight into the problem and introduce new modelling approaches for solving real world problems.

2. The solution methods developed will facilitate high quality solution and help industry improve efficiency and productivity.

3. The solution methods developed will be helpful in solving other discrete optimization problems with similar structures.

Intermodal and LTL transportation companies are faced with decision problems at all levels. This dissertation provides quantitative model development and solution algorithms for these problems. Intermodal and LTL transportation firms must make:

1. Strategic decisions about capital investments such as buildings, facilities and

equipment for loading, unloading and sorting activities. Transportation capacities must be acquired in the form of either an owned or rented fleet.

2. Tactical capacity planning decisions, such as whether to purchase trucks for various linehaul links or to acquire capacity from rental agreements. The cost of acquiring capacity on an emergency or expedited basis costs much more than the regular price; therefore, appropriate capacity planning can help reduce the cost of emergency capacity acquisition.

3. Operational decisions, such as truck-linehaul assignments and commodity-truck-linehaul assignments and other operational constraints.

The decision problems at the strategic, tactical and operational levels are interrelated. Decisions made at the strategic level provide the resources/information required at the tactical and operational level, and, similarly, decisions made at the strategic and tactical levels affect operational level decisions. In this dissertation we investigate the decision problems at all three levels-operational, tactical and strategic. Our approach for each decision problem is to first articulate the scope of the problem, and then conduct a literature survey followed by the development of dedicated models and efficient solution methods.

All three levels of problems are defined on the same network which can be described as follows. We are given a network with multi-commodity flows where each commodity is defined by its unique pair of origin and destination nodes and a known required flow amount. The system is operated in such a way that the commodities are collected and consolidated into truckloads at consolidation centers, a linehaul transfer takes place for the consolidated loads, which are deconsolidated at deconsolidation centers, and from there, the commodities are shipped to their final destinations.

In addition to the network and commodity flows, in ONDP we are also given a fleet of trucks . Additionally, we allow direct shipments between origin and destination nodes since this is preferred when the origin and destination nodes of a commodity are relatively close, and, thus, consolidation does not make economical sense. The decisions to be made in ONDP include 1) the assignment of trucks to linehaul transfer links, 2) the assignment of commodities to a truckload shipment established on transfer links, and 3) the identification of commodities that are to be shipped directly.

In case of TNDP, we consider longer planning horizons and we are no longer given the truck fleet, plus we do not consider direct shipment. The decisions concerned with TNDP are 1) the connections and capacities in terms of the number of *truckload trips* between the consolidation and deconsolidation centers, and 2) the assignment of commodities to consolidation and deconsolidation centers and, in turn, to transfer links. The costs in the system include collection costs, linehaul transfer costs and distribution costs.

The strategic planning problem SNDP extends the TNDP to include location and capacity decisions related to centers which are capital intensive decisions. In SNDP, we allow the multiple allocation of commodities to the centers and, thereby, to the transfer links.

The literature relevant to the problems addressed in this dissertation can be classified in four categories, namely hub location, network design, service network design and facility location. Since our problem concerns the design of a distribution network and involves consolidation and deconsolidation activities, the hub location literature is naturally relevant to us, and since we are interested in routing commodities over a network, the general area of network design is also relevant. Furthermore, due to the specific application area, studies in logistics service network design and load planning are also within our scope. Finally, since each linehaul link can also be considered as a facility, our problem is related to capacitated facility location problems. In Chapter II, we presented a detailed survey of the literature in the above mentioned areas. Below we summarize the distinction between our problems and the problems studied in the literature.

Our problems, TNDP and ONDP, do not involve hub (or center) location decisions. Further, in all three problems ONDP, TNDP and SNDP, we consider explicit commodity-based routing decisions, which is not the case in hub location problems. Therefore, despite operational similarities, the problems considered in this dissertation have fundamental differences from the hub location problems , which hinder the efficient use of the modelling and solution approaches devised for these problems.

In the context of network design, our problems distinguish themselves from the general network design problems in the sense that they consider the consolidation and de-consolidation activities explicitly; in addition, the transportation costs on the transfer links from consolidation centers to de-consolidation centers is not linear, but a step function of the quantity being transferred. Further, as opposed to network design problems, our problem has a structure that requires use of exactly three arcs (or in case of ONDP, direct shipment) which is not the case in general network design problems.

We presented a transformation to construct an SSCFLP equivalent of ONDP in Chapter III, page 46. Similarly, TNDP is nothing but a SSCFLP with staircase capacities. In this context, we presented a literature review of facility location problems. We also showed that the SSCFLP equivalent of ONDP is prohibitively large for the state-of-the-art methods to solve. TNDP is a SSCFLP with staircase cost and has not been solved in the literature. We presented an extensive review of the SSCFLP, its applications and solution methods.

We developed a compact binary program formulation for ONDP. The model considers the transportation *economies-of-scale* in a simple yet effective manner. For the purpose of developing efficient solution algorithms, we first proposed four compound neighborhood functions where each has two main components, level- and content-change, with the latter being based on various schemes of combining simple neighborhood functions such as move and exchange. Given the complexity of an efficient solution representation required in a heuristic framework, our compound neighborhood functions enable us to search the solution space efficiently using a branching strategy that we introduce. In addition, the two components provide the means for efficiently incorporating intensification and diversification characteristics into the heuristic search algorithms. We developed three heuristic algorithms based on local search, simulated annealing and tabu search where each can employ each of the four compound neighborhood functions and the proposed branching strategy, thus giving rise to twelve different approaches. We performed extensive computational experiments and obtained results illustrating the relative efficiency and effectiveness of our compound neighborhood functions and heuristic algorithms.

We developed an integer programming formulation and methods for obtaining lower and upper bounds for the TNDP. We modified the compound neighborhood search procedures to adapt to the tactical problem and developed a Simulated Annealing-based heuristic algorithm to provide a feasible solution, which is an upper bound. As opposed to one-exchange and one-move based neighborhoods, in TNDP, we developed two-exchange and two-move based simple neighborhoods that provides good quality solution in TNDP, despite the fact that there are fewer numbers of simple content-change neighborhoods (two as opposed to 5 in ONDP). In a Simulated Annealing-based metaheuristic, we implemented a sophisticated bias tree search strategy that proves to be very efficient in guiding the search in a direction with potentially good solutions. For finding a lower bound, we proposed a Lagrangian relaxation-based solution method and theoretically showed it to be tighter than the LP relaxation lower bound. We also suggested a variable relaxation idea to speed up the computation of the Lagrangian relaxation-based lower bound without compromising the solution quality. Finally, we developed a Lagrangian heuristic framework that utilizes the upper bound found by a Simulated Annealing-based metaheuristic to update the Lagrange multipliers using sub-gradient optimization. The Lagrangian heuristic framework also utilizes the lower bound solution as input to the Simulated Annealing method to find even better upper bounds. The computational study confirms the superiority of the Lagrangian relaxation-based lower bound over LP-based lower bounds.

We presented a mixed integer programming formulation for SNDP that has a structure in which, for fixed capacities, the subproblem reduces to the transportation problem, and the integer program involving only the integer variables is relatively small. Such a structure motivates the Benders decomposition method as a solution methodology. We developed a Benders decomposition-based solution method for solving SNDP. We tested our solution methodology on two sets of data, first generated randomly using the parameters similar to the ONDP and TNDP and second generated using a large parcel company's data. We found that the Benders-based solution method performed consistently well for small as well as comparatively larger problems.

VI.1. Contributions

Major contributions of this dissertation are:

- *Holistic Approach for Intermodal and LTL Network Design:* We provide quantitative models for strategic, tactical and operational levels decision problems faced by intermodal and LTL transportation companies. The solution of the strategic problem provides the resources/infrastructure for the tactical problem and insures efficient planning at the tactical and operational level. The solution to the tactical planning problem is crucial in enabling the execution of operational decisions. Therefore, the dissertation provides complete and insightful models and their solutions to the intermodal and LTL transportation companies.

- *Compound Neighborhood Search Based Solution Framework:* The compound neighborhood search method developed in this dissertation can be used to solve many other problems. Its versatility is demonstrated in the dissertation itself when the framework developed in ONDP was able to provide good quality solutions for TNDP after adapting the ingredient components to the problem structure of TNDP. We believe that this method can be beneficial in solving other combinatorial optimization problems as long as one can find a computationally inexpensive way to generate the neighborhood solutions and an efficient way of compounding them.

- *SSCFLP with Staircase Capacity:* The single source capacitated facility problem with staircase capacity has not been solved in the literature. Moreover, the application we are considering, the equivalent SSCFLP are much larger than the traditional location problems. Therefore, this dissertation contributes in developing efficient solution approaches for SSCFLP by combining metaheursitic-based upper bounds and Lagrangian relaxation- based lower bounds through a Lagrangian heuristics framework.

- *Benders Decomposition Based Approach:* The mathematical model for the SNDP has a structure suitable for Benders decomposition-based solution methods. The dissertation extends the application of the Benders method to a new problem successfully and also extends the strong cut generation to this new problem successfully.

VI.2. Foundation for Future Research

The models and solution methods developed in this dissertation can be extended in the future to consider generalizations and complexities as described below:

1. *Generalized cost functions:* Although the cost function in our study involves step function, all of our models and solution methods can be modified for other piecewise linear cost functions as well. In a piecewise linear cost function, each linear segment has a fixed cost called the intercept, a variable cost which is the slope of the linear segment, and the upper and lower breakpoints. The step function is a special piecewise cost function where the slopes are zero for all segments. Similarly, the case where all the segments have same intercept (possibly zero) but different slopes is known as all unit discount. Using any one of the three textbook models (Croxton et al., 2003a) for piecewise cost functions; namely incremental, multiple choice or convex combination, our problem formulations can also be extended for general piecewise linear cost functions. Croxton et al. (2003a) compare these three formulations of the piecewise nonlinear costs and shows their equivalence with respect to LP relaxations. By modifying objective function evaluation, the compound neighborhood function can also be utilized for general piecewise linear cost functions. The Lagrangian heuristic and Benders decomposition-based method can also be applied to formulations extended to model general piecewise linear cost functions. Some studies considering piecewise linear concave costs include Balakrishnan and Graves (1989),

Amiri and Pirkul (1997), Croxton et al. (2003b) and Muriel and Munshi (2003).

2. *Managerial Insight:* The tactical and strategic problems are based on estimated demands which may not always be realized. The models developed in this dissertation may provide managerial insight. Assuming that the fluctuations in the demand realization are not major, the model provides quick estimate of change in the total cost by simple capacity adjustments (removing the unused capacity, and installing additional capacity on links where required) and evaluation of the resulting solution. For large variations, the manager can re-solve the problem.

3. *Time window constraint:* The transportation industry often faces time window constraints. Increased competition and customer's expectations for quality service make time window constraints all the more important. Our models and solution approaches can be extended to include time window constraints.

4. *Generalized cost functions:* In practice, logistics service providers may offer quantity or other types of contract based discounts. Sometimes a single tractor is used to pull up to three trailers and this practice is called tandem trailer. Discounts and tandem trailers may give rise to complex cost functions. Future extensions of our models and solution approaches may address complex cost functions.

5. *Applying solution methods to other problem domains:* The solution methods that we have developed in this dissertation may also be used in other combinatorial optimization problems. For example, the Lagrangian relaxation approach may be used in location problems whereas the heuristics may be used in generalized assignment problem.

6. *Other issues in LTL and intermodal transportation:*There are several other im-

portant issues relevant to LTL and intermodal transportation, such as repositioning of empties, maintenance scheduling and driver turnover. These issues may have been addressed individually in the literature. However, inclusion of these issues along with consolidation and routing decisions is a potential future extension of our models.

REFERENCES

Aarts, E., J. H. M. Korst, P. J. M. van Laarhoven. 1997. Simulated annealing. E. Aarts, J. K. Lenstra, eds., *Local Search in Combinatorial Optimization*. John Wiley and Sons, New York, NY 91–120.

Agar, M. C., S. Salhi. 1998. Lagrangean heuristics applied to a variety of large capacitated plant location problems. *The Journal of the Operational Research Society* **49**(10) 1072–1084.

Agarwal, Y. K. 2002. Design of capacitated multicommodity networks with multiple facilities. *Operations Research* **50**(2) 333–344.

Ahuja, R. K., T. L. Magnanti, J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Englewood Cliffs, NJ.

Ahuja, R. K., J. B. Orlin, S. Pallottino, M. P. Scaparra, M. G. Scutella. 2004. A multi-exchange heuristic for the single-source capacitated facility location problem. *Management Science* **50**(6) 749–760.

Akinc, U., B. M. Khumawala. 1977. An effcient branch and bound algorithm for the capacitated warehouse location problem. *Management Science* **23** 585–194.

Amiri, A., H. Pirkul. 1997. New formulation and relaxation to solve a concave-cost network flow problem. *Journal of Operational Research Society* **48** 278–287.

Armacost, A. P., C. Barnhart, K. A. Ware. 2002. Composite variable formulations for express shipment service network design. *Transportation Science* **36**(1) 1–20.

Armacost, A. P., C. Barnhart, K. A. Ware, A. M. Wilson. 2004. UPS optimizes its air network. *Interfaces* **34**(1) 15–25.

Atamtürk, A. 2002a. On capacitated network design cut-set polyhedra. *Mathematical Programming, Series B* **92** 425–437.

Atamtürk, A. 2002b. On splittable and unsplittable flow capacitated network design arc-set polyhedra. *Mathematical Programming, Series A* **92** 315–333.

Balakrishnan, A., S. C. Graves. 1989. A composite algorithm for a concave-cost network flow problem. *Networks* **19** 175–202.

Balakrishnan, A., T. L. Magnanti, R. T. Wong. 1989. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research* **37**(5) 716–740.

Barcelo, J., J. Casanovas. 1984. A heuristic lagrangean algorithm for the capacitated plant location problem. *European Journal of Operational Research* **15**(2) 212–226.

Barnhart, C., C. A. Hane, P. H. Vance. 2000. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research* **48**(2) 318–326.

Barnhart, C., N. Krishnan, D. Kim, K. Ware. 2002. Network design for express shipment delivery. *Computational Optimization and Applications* **21**(3) 239–262.

Barnhart, C., R. R. Schneur. 1996. Air network design for express shipment service. *Operations Research* **44**(6) 852–863.

Beasley, J. E. 1993. Lagrangean heuristics for location problems. *European Journal of Operational Research* **65**(3) 383–399.

Bienstock, D., O. Günlük. 1996. Capacitated network design-polyhedral structure and computation. *Journal on Computing* **8** 243–259.

Bowersox, D. J., D. J. Closs, M. B. Cooper. 2002. *Supply Chain Logistical Management*. The McGraw-Hill Companies, Inc., New York, NY.

Bowersox, D.J., D.J. Closs, O.K. Helferich. 1986. *Logistical Management: A Systems Integration of Physical Distribution, Manufacturing Support, and Materials Procurement*. Macmillan, New York.

Braklow, J. W., W. W. Graham, S. M. Hassler, K. E. Peck, W. B. Powell. 1992. Interactive optimization improves service and performance for Yellow Freight System. *Interfaces* **22**(1) 147–172.

Büdenbender, K., T. Grünert, H-J. Sebastian. 2000. A hybrid tabu search/branch-and-bound algorithm for the direct flight network design problem. *Transportation Science* **34**(4) 364–380.

Caldwell, H., J. Sedor. 2002. The freight story: A national perspective on enhancing freight transportation. Tech. rep. USDOT Washington DC.

Campbell, J. F. 1994. A survey of network hub location. *Studies in Locational Analysis* **6** 31–49.

Campbell, J. F. 2005. Strategic network design for motor carriers. A. E. D. T. Langevin, ed., *Logistics Systems: Design and Optimization*. Springer, New York 245–278.

Campbell, J. F., A. T. Ernst, M. Krishnamoorthy. 2002. Hub location problems. Z. Drezner, H. W. Hamacher, eds., *Facility Location: Applications and Theory*. Springer-Verlag, New York, NY 373–408.

Campbell, JF, AT Ernst, M. Krishnamoorthy. 2005a. Hub arc location problems: Part I–introduction and results. *Management Science* **51**(10) 1540–1555.

Campbell, JF, AT Ernst, M. Krishnamoorthy. 2005b. Hub Arc Location Problems: Part II–Formulations and Optimal Algorithms. *Management Science* **51**(10) 1556–1571.

Cornuejols, G., M. L. Fisher, G. L. Nemhauser. 1990. The uncapacitated facility location problem. P. B. Mirchandani, R. L. Francis, eds., *In Discrete Location Theory*. John Wiley and Sons 119–171.

Correia, I., M. E. Captivo. 2006. Bounds for the single source modular capacitated plant location problem. *Computers and Operations Research* **33**(10) 2991–3003.

Cortinhal, M. J., M. E. Captivo. 2003. Upper and lower bounds for the single source capacitated location problem. *European Journal of Operational Research* **151**(2) 333–351.

Costa, A.M. 2005. A survey on benders decomposition applied to fixed-charge network design problems. *Computers and Operations Research* **32**(6) 1429–1450.

Crainic, T. G. 1999. Long-haul freight transportation. R. W. Hall, ed., *Handbook of Transportation Science*. Kluwer, Boston MA 451–516.

Crainic, T. G. 2000. Service network design in freight transportation. *European Journal of Operational Research* **122** 272–288.

Crainic, T. G., M. Florian, J-E. Leal. 1990. A model for the strategic planning of national freight transportation by rail. *Transportation Science* **24**(1) 1–24.

Crainic, T. G., A. Frangioni, B. Gendron. 2001. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics* **112** 73–99.

Crainic, T. G., M. Gendreau, J. M. Farvolden. 2000. A simplex-based tabu search method for capacitated network design. *Journal on Computing* **12**(3) 223–236.

Crainic, T. G., K. H. Kim. 2005. *Intermodal Transportation*. Centre de recherche sur les transports, Universite de Montreal, Quebec, Canada. Extracted from "http://www.crt.umontreal.ca/ theo/mba8L91/CrainicKimIntermodal.pdf".

Crainic, T. G., G. Laporte. 1997. Planning models for freight transportation. *European Journal of Operational Research* **97** 409–438.

Crainic, T. G., J-M. Rousseau. 1986. Multicommofity, multimode freight transportation: a general modeling and algorithmic framework for the service network design problem. *Transportation Research: Part B–Methodological* **20**(3) 225–242.

Croxton, K. L., B. Gendron, T. L. Magnanti. 2003a. A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science* **49** 1268–1273.

Croxton, K. L., B. Gendron, T. L. Magnanti. 2003b. Models and methods for merge-in-transit operations. *Transportation Science* **37**(1) 1–22.

Daskin, M. S. 1995. *Network and Discrete Location: Models, Aplications and Algorithms*. John Wiley and Sons, Inc., New York, NY.

Davis, P. S., T. L. Ray. 1969. A branch-bound algorithm for the capacitated facilities location problem. *Naval Research Logistics Quarterly* **16**(3) 331–44.

Dell'Amico, M., Maffioli F., S. Martello, eds. 1997. *Annonated Bibliogrphies in Combinatorial Optimization*. John Wiley and Sons, New York, NY.

Delmaire, H., J. A. Diaz, E. Fernandez, M. Ortega. 1999. Reactive GRASP and Tabu Search based heuristics for the single source capacitated plant location problem. *INFOR* **37** 194–225.

Delorme, L., J. Roy, J-M. Rousseau. 1987. Motor-carriers operations planning: a state of the art. L. Bianco, A. La Bella, eds., *Freight Transport Planning and Logistics*. Springer-Verlag, Berlin 510–545.

Diaz, J. A., E. Fernandez. 2002. A branch-and-price algorithm for the single source capacitated plant location problem. *Journal of the Operational Research Society* **53** 728–740.

Eckstein, J., Y. Sheffi. 1987. Optimization of group line-haul operations for motor carriers using twin trailers. *Transportation Research Record* **1120** 12–23.

Erlenkotter, D. 1978. A dual-based procedure for uncapacitated facility location. *Operations Research* **26**(6) 992–1009.

Ernst, A. T., M. Krishnamoorthy. 1999. Solution algorithms for the capacitated single allocation hub location problem. *Annals of Operations Research* **86** 141–159.

Farvolden, J. M., W. B. Powell. 1994. Subgradient methods for the service network design problem. *Transportation Science* **28**(3) 256–272.

FHWA. 2006a. Freight Analysis,U.S. Department of Transportation, Federal Highway Administration , report numebr FHWA-OP-04-054. Extracted from www.fhwa.dot.gov/operations on April 30 2007.

FHWA. 2006b. Freight Facts and Figures, U.S. Department of Transportation, Federal Highway Administration , report numebr, FHWA-HOP-07-033. Extracted from http://www.ops.fhwa.dot.gov/freight on April 30 2007.

Francis, R. L., L. F. McGinnis, J. A. White. 1992. *Facility Layout and Location: An Analytical Approach.*. Prentice-Hall, Englewood Cliffs, NJ.

FWHA. 1998. U.s. Freight: Economy in motion. *Report No. FHWA-PL-98-034* 1–24U.S. Department of Transportation, Federal Highway Administration.

Gendron, B., T. G. Crainic, A. Frangioni. 1998. Multicommodity capacitated network design. B. Sansó, P. Soriano, eds., *Telecommunications Network Planning*. Kluwer, Berlin 1–19. Extracted from "citeseer.ist.psu.edu/gendron97multicommodity.html".

Geoffrion, A. M., G. W. Graves. 1996. Multicommodity distribution system design by benders decomposition. *Management Science* **20** 822–844.

Ghiani, G., G. Laporte, R. Musmanno. 2004. *Introduction to Logistics Systems Planning and Control*. John Wiley and Sons, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England.

Glover, F., M. Laguna. 1997. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA.

Grünert, T., H-J. Sebastian. 2000. Planning models for long-haul operations of postal and express shipment companies. *European Journal of Operational Research* **122**(2) 289–309.

Guelat, J., M. Florian, T. G. Crainic. 1990. A multimode multiproduct network assignment model for strategic planning of freight networks. *Transportation Science* **24**(1) 25–39.

Hindi, K. S., K. Pienkosz. 1999. Efficient solution of large scale, single-source, capacitated plant location problems. *The Journal of the Operational Research Society* **50**(3) 268–274.

Holmberg, K., M. Ronnqvist, D. Yuan. 1999. An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research* **113**(3) 544–559.

Holmberg, K., D. Yuan. 2000. A lagrangean heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research* **48**(3) 461–481.

Klincewicz, J. G., H. Luss. 1986. A lagrangian relaxation heuristic for capacitated facility location with single-source constraints. *The Journal of the Operational Research Society* **37**(5) 495–500.

Kuby, M. J., R. G. Gray. 1993. The hub network design problem with stopovers and feeders: the case of Federal Express. *Transportation Research Part A: Policy and Practice* **27**(1) 1–12.

Lamar, B. W., Y. Sheffi. 1987. An implicit enumeration method for LTL network design. *Transportation Research Record* **1120** 1–11.

Lamar, B. W., Y. Sheffi, W. B. Powell. 1990. A capacity improvement lower bound for fixed charge network design problem. *Operations Research* **38**(4) 704–710.

Lapierre, S. D., A. B. Ruiz, P. Soriano. 2004. Designing distribution networks: Formulations and solution heuristic. *Transportation Science* **38**(2) 174–187.

Lemaréchal, C. 1989. Nondifferentiable optimization. G. L. Nemhauser, A. H. G. Rinnoy Kan, M. J. Todd, eds., *Handbooks in Operations Research and Management Science, Volume 1: Optimization*. Elsevier, Amsterdam Netherlands 529–572.

Leung, J. M. Y., T. L. Magnanti, V. Singhal. 1990. Routing in point-to-point delivery systems: formulations and solution heuristics. *Transportation Science* **24**(4) 245–260.

Love, R.F., J.G. Morris, G.O. Wesolowsky. 1988. *Facilities Location. Models and Methods*. North Holland Publishing Company, New York, NY, USA.

Magnanti, T. L., P. Mirchandani, R. Vachani. 1993. The convex hull of two core capacitated network design problems. *Mathematical Programming* **60** 233–250.

Magnanti, T. L., P. Mirchandani, R. Vachani. 1995. Modeling and solving the two-facility capacitated network loading problem. *Operations Research* **43**(1) 142–157.

Magnanti, T. L., R. T. Wong. 1984. Network design and transportation planning: Models and algorithms. *Transportation Science* **18** 1–55.

Magnanti, TL, RT Wong. 1981. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research* **29**(3) 464–484.

Magnanti, T.L., R.T. Wong. 1986. Decomposition methods for facility location problems. *working paper* Extracted from http://hdl.handle.net/1721.1/5128.

Minoux, M. 1989. Network synthesis and optimum network design problems: Models, solution methods and applications. *Network* **19** 313–360.

Minoux, M. 2001. Discrete cost multicommodity network optimization problems and exact solution methods. *Annals of Operations Research* **106** 19–46.

Muriel, A., F. N. Munshi. 2003. Capacitated multicommodity network flow problems with piecewise linear concave costs. *IIE Transactions* **36** 683–696.

Neebe, A. W., M. R. Rao. 1983. An algorithm for the fixed-charge assigning users to sources problem. *The Journal of the Operational Research Society* **34**(11) 1107–1113.

O'Kelly, M. E., H. J. Miller. 1994. The hub network design problem: A review and synthesis. *Journal of Transport Geography* **2**(1) 31–40.

Pirkul, H. 1987. Efficient algorithms for the capacitated concentrator location problem. *Computers and Operations Research* **14**(3) 197–208.

Powell, W. B. 1986. A local improvement heuristics for the design of Less-than-Truckload motor carrier networks. *Transportation Science* **20** 246–257.

Powell, W. B., Y. Sheffi. 1983. The load planning problem of motor carriers: problem description and a proposed solution approach. *Transportation Research Part A: Policy and Practice* **17** 471–480.

Powell, W. B., Y. Sheffi. 1989. Design and implementation of an interactive optimization system for network design in the motor carrier industry. *Operations Research* **37**(1) 12–29.

Rardin, R. L. 1982. Tighter relaxations of fixed charge network flow problems. Tech. Rep. J–82–3 School of Industrial and Systems Engineering, Georgia Institute of Technology Atlanta GA.

Rardin, R. L., U. Choe. 1979. Tighter relaxations of fixed charge network flow problems. Tech. Rep. J–79–18 School of Industrial and Systems Engineering, Georgia Institute of Technology Atlanta GA.

Roy, T. J. V. 1986. A cross decomposition algorithm for capacitated facility location. *Operations Research* **34**(1) 145–163.

Sait, S. M., H. Youssef. 1999. *Iterative Computer Algorithms with Applications in Engineering: Solving Combinatorial Optimization Problems*. IEEE Computer Society, Los Alamitos, CA.

Sridharan, R. 1993. A lagrangian heuristic for the capacitated plant location problem with single source constraints. *European Journal of Operational Research* **66**(3) 305–312.

Sridharan, R. 1995. The capacitated plant location problem. *European Journal of Operational Research* **87**(2) 203–213.

Üster, H., N. Maheshwari. 2007. Strategic network design for multi-zone truckload shipments. *IIE Transactions* **39**(2) 177–189.

Wentges, P. 1996. Accelerating Benders' decomposition for the capacitated facility location problem. *Mathematical Methods of Operations Research (ZOR)* **44**(2) 267–290.

Wolsey, L. A. 1998. *Integer Programming*. John Wiley and Sons, New York, NY.

VITA

Homarjun Agrahari received his Bachelor of Technology Degree in Mechanical Engineering from mechanical engineering in 2001 and Master of Technology in Mechanical Engineering with specialization in Computer Integrated Manufacturing in 2001, both from Indian Institute of Technology, Bombay. After working for two years in the industry, In fall of 2003, he joined the Industrial Engineering department for doctoral studies. He is interested in developing mathematical models and efficient exact and heuristic solution methodologies for discrete optimization problems arising in industries such as transportation and parcel services. He received his Ph.D. and joined BNSF Railway, Fort Worth in 2007.

Homarjun can be reached at his work address:

BNSF Railway Head Quarters

2650 Lou Menk Dr.

Fort Worth, TX 76131